

TUGAS AKHIR - TE 141599

PENGENDALIAN KEPADATAN DALAM MULTIMEDIA STREAMING DENGAN ACTIVE QUEUE MANAGEMENT

Ilham Fahmi Kurniawan NRP 2211 100 034

Dosen Pembimbing Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D. Nurlita Gamayanti, S.T., M.T.

JURUSAN TEKNIK ELEKTRO Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya 2016



FINAL PROJECT - TE 141599

CONGESTION CONTROL IN MULTIMEDIA STREAMING WITH ACTIVE QUEUE MANAGEMENT

Ilham Fahmi Kurniawan NRP 2211 100 034

Supervisors Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D. Nurlita Gamayanti, S.T., M.T.

DEPARTMENT OF ELECTRICAL ENGINEERING Faculty of Industrial Technology Institut Teknologi Sepuluh Nopember Surabaya 2015

PENGENDALIAN KEPADATAN DALAM MULTIMEDIA STREAMING DENGAN ACTIVE QUEUE MANAGEMENT

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk Memperoleh Gelar Sarjana Teknik Elektro Pada Bidang Studi Sistem Pengaturan

Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui,

Dosen Pembimbing I,

ein.

Dosen Pembimbing II,

Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D. NIP. 1955 01 23 1980 03 1002 Nurlita Gamayanti, S.T., M.T. NIP. 1978 12 01 2002 12 2002

SURABAYA
JUNI, 2016
NIK ELEKTRO

PENGENDALIAN KEPADATAN DALAM MULTIMEDIA STREAMING DENGAN ACTIVE OUEUE MANAGEMENT

Ilham Fahmi Kurniawan 2211 100 034

Dosen Pembimbing I : Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.

Dosen Pembimbing II : Nurlita Gamayanti, S.T., M.T.

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 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 AOM dengan algoritma Drop-Tail untuk menemukan nilai OoS 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.

CONGESTION CONTROL IN MULTIMEDIA STREAMING WITH ACTIVE QUEUE MANAGEMENT

Ilham Fahmi Kurniawan 2211 100 034

Supervisor I : Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.

Supervisor II : Nurlita Gamayanti, S.T., M.T.

ABSTRACT

Multimedia Streaming that increasingly more and more used by internet users led to overcrowding on the network the internet. Generally, Drop-Tail Queue Management algorithm used to set the system queues in the buffer in the case of streaming multimedia. These are the causes a pretty significant distortion. Active Queue Management (AQM) is a method of regulating the entry of data packets into a buffer as a queue system in computer networks. In this final task, AQM is used as a solution to regulate the length of the queue and the time delay in the buffer, which is where the data packets. In a buffer, the system queues the data package processing will be subject to bring up a virtual queue as a media processing while in the buffer. The research carried out using scenarios from multimedia stream which has been modeled into mathematical models and then compare Quality-of-Service (QoS) in the form of average waiting time in the system and packet-loss value of AQM with Drop-Tail algorithm to find the best QoS values. The result concluse that system with AOM is better than system with Drop-Tail algorithm, proof with number of packet-loss less than 15% and faster delay.

Keyword: Active Queue Management, Buffer, Drop-Tail, Multimedia Streaming, Quality-of-Service, Queueing System.

DAFTAR ISI

	AN JUDUL	
PERNYA'	TAAN KEASLIAN	V
LEMBAR	PENGESAHAN	. vii
ABSTRA	K	ix
ABSTRAC	CT	xi
KATA PE	NGANTAR	xiii
DAFTAR	ISI	. xv
DAFTAR	GAMBAR	xix
	TABEL	
BAB 1 PE	NDAHULUAN	1
1.1	Latar Belakang	1
1.2	Permasalahan	2
1.3	Batasan Masalah	2
1.4	Tujuan	2
1.5	Metodologi Penelitian	3
1.6	Sistematika Pembahasan	4
1.7	Relevansi	5
BAB 2 TE	ORI PENUNJANG	7
2.1	Sistem Antrian	7
2.1.1	Tujuan sistem antrian	7
2.1.2	Notasi kendall	7
2.1.3	Model sistem antrian markovian	8
2.1.4	Little's formula	. 10
2.1.5	Performa sistem antrian	. 10
2.2	Multimedia Streaming	. 16
2.2.1	Metode pengiriman data pada multimedia streaming	. 18
2.2.2	Proses multimedia streaming di buffer	. 18
2.3	Algoritma Active Queue Management (AQM)	. 20
2.4	Algoritma Drop-Tail Queue Management	. 21
2.5	Quality-of-Service (QoS)	. 21
2.5.1	Jitter	. 22
252	Randwidth	22

2.5.3	Latency	22
2.5.4	Packet-loss	23
2.5.5	Throughput	23
BAB 3 PE	RANCANGAN SISTEM	25
3.1	Gambaran Umum Sistem	25
3.2	Perancangan Model Matematis Sistem	27
3.2.1	Fungsi objektif	27
3.2.2	Fungsi kendala	27
3.3	Perancangan Simulasi	29
3.3.1	Waktu antar kedatangan	29
3.3.2	Kapasitas antrian	30
3.3.3	Waktu pelayanan	31
3.3.4	Waktu time-out paket data di dalam sistem	31
3.3.5	Jumlah server	32
3.3.6	Proses multimedia streaming pada simulasi	32
BAB 4 SI	MULASI SISTEM	35
4.1	Spesifikasi Server dan Pengolahan Data	35
4.1.1	Spesifikasi server Kirby	35
4.1.2	Latency	36
4.1.3	Memory buffer	37
4.1.4	Load	38
4.1.4	Incoming user requests	39
4.2	Simulasi Sistem Antrian dengan Algoritma AQM	40
4.2.1	Keterangan parameter simulasi sistem	40
4.2.2	Penerapan algoritma AQM	41
4.3	Simulasi sistem antrian dengan algoritma Drop-Tail	43
BAB 5 PE	NGUJIAN SISTEM	45
5.1	Tujuan Pengujian	45
5.2	Pengujian QoS Sistem Antrian	45
5.2.1	Pengujian simulasi algoritma AQM	46
5.2.2	8-J8 F	
5.2.3		
BAB 6 PE	NUTUP	53
6.1	Kesimpulan	53

6.2	Saran	53
DAFTAR	PUSTAKA	55
LAMPIR	AN A	57
A.1	Script pada Simulasi Algoritma AQM	57
A.2.	Tampilan Simulasi Algoritma AQM di SimuLink	58
LAMPIR	AN B	61
B.1	Script pada Simulasi Algoritma Drop-Tail	61
B.2	Tampilan Simulasi Algoritma Drop-Tail di SimuLink	62
LAMPIR	AN C	65
C.1	Hasil Simulasi Algoritma AQM	65
C.2	Hasil Simulasi Algoritma Drop-Tail	69
RIWAYA	AT HIDUP	73

DAFTAR GAMBAR

Gambar 2.1. Diagram transisi proses kelahiran dan kematian	8
Gambar 2.2. Diagram transisi antrian single server	9
Gambar 2.3. Diagram transisi antrian multi server	9
Gambar 2.4. Diagram transisi antrian multi server kapasitas terbatas.	10
Gambar 2.5. Ilustrasi multimedia streaming	17
Gambar 2.6. Ilustrasi pengiriman data video streaming	18
Gambar 2.7. Proses yang terjadi pada video streaming	18
Gambar 2.8. Skema proses multimedia streaming	
Gambar 2.9. Ilustrasi pengiriman data multi stream video [1]	20
Gambar 2.10. Ilustrasi pengimplementasian algortima Drop-Tail [8].	21
Gambar 3.1. Proses streaming yang dimodelkan dan disimulasikan	
Gambar 3.2. Sistem antrian pada buffer [1]	26
Gambar 3.3. Gambar block parameters time-based entity generator	29
Gambar 3.4. Gambar kotak dialog untuk memasukkan tipe entity	29
Gambar 3.5. Gambar kotak dialog untuk memasukkan nilai input	30
Gambar 3.6. Gambar block parameters FIFO queue	30
Gambar 3.7. Gambar kotak dialog untuk kapasitas maksimal	30
Gambar 3.8. Gambar block parameters single server	31
Gambar 3.9. Gambar kotak dialog untuk memasukkan nilai input	31
Gambar 3.10. Gambar block parameters setting time-out	31
Gambar 3.11. Gambar kotak dialog untuk delay maksimal	32
Gambar 3.12. Peran read file pada simulasi sistem	33
Gambar 3.13. Peran write buffer pada simulasi sistem	33
Gambar 3.14. Peran send buffer pada simulasi sistem	34
Gambar 4.1. Grafik rata-rata dan maksimal latency	36
Gambar 4.2. Grafik rata-rata dan maksimal memory buffer	38
Gambar 4.3. Grafik nilai rata-rata beban kerja server	39
Gambar 4.4. Penggunaan algoritma mencari antrian terpendek	42
Gambar 4.5. Gambar pengaturan waktu time-out pada server	42
Gambar 4.6. Penerapan script untuk lock-out dan full queue	
Gambar 5.1. Grafik perbandingan packet-loss (K=60)	48
Gambar 5.2. Grafik perbandingan packet-loss (K=108)	48

Gambar 5.3. Grafik perbandingan total delay (K=60)	49
Gambar 5.4. Grafik perbandingan total delay (K=108)	50
Gambar 5.5. Grafik perbandingan jitter (K=60)	50
Gambar 5.6. Grafik perbandingan jitter (K=108)	51

DAFTAR TABEL

22 23
23
35
36
37
38
40
41
46
46
47
47

BAB 1 PENDAHULUAN

1.1 Latar Belakang

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 di atas 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 spefikasi dari jalur transmisi video dengan jaringan lainnya, dirumuskan sebagai suatu kebutuhan khusus untuk AQM dalam kasus peningkatan *Quality-of-Service* (QoS) dari *scalable video streaming*.

Dalam tugas akhir ini, diusulkan sistem AQM berbasis antrian untuk mengurangi rata-rata panjang antrian dan waktu tunda antrian pada *video streaming* sementara rata-rata *packet loss* tidak berkurang sampai dengan batas tertentu yang diatur oleh penyedia jasa layanan dan menjamin QoS berada pada titik yang dapat diterima. Dalam menjalankan

penelitian yang diusulkan, penulis merancang model sistem antrian di *buffer* dengan bantuan referensi yang telah ada.

1.2 Permasalahan

Pada tugas akhir kali ini yang menjadi permasalahan utama adalah menyelesaikan permasalahan antrian pada *buffer* dengan algoritma AQM sehingga QoS pada penerapan algoritma AQM dapat dibandingkan dengan QoS pada penerapan algoritma Drop-Tail.

1.3 Batasan Masalah

Hal – hal yang menjadi batasan dalam penelitian ini adalah sebagai berikut.

- 1. Jenis *file* data yang digunakan dalam penelitian ini adalah *file* video.
- 2. Selama proses *multimedia streaming* diasumsikan antrian hanya terjadi pada *buffer*, yang berarti *delay* yang terjadi di selain *buffer* selama proses *multimedia streaming* bernilai 0.
- 3. Ukuran data, kompleksitas data, dan jumlah data pada setiap paket data dalam penilitian ini adalah identik.
- 4. Kapasitas antrian dan laju pelayanan pada setiap *server* bernilai sama atau identik.
- 5. Jumlah *server* dalam sistem antrian yang dirancang ditentukan sesuai kebutuhan penelitian.
- 6. Kualitas dan sinyal jaringan diasumsikan selalu stabil dan dalam kondisi terbaik, yang berarti tidak ada *transportation delay* atau waktu tunggu antar proses.

1.4 Tujuan

Penelitian ini bertujuan guna menerapkan algoritma AQM berbasis antrian agar mendapatkan kualitas layanan, yang meliputi *jitter*, total *delay* dan *packet-loss*, yang lebih baik pada *multimedia streaming* dengan penerapan algoritma tersebut pada model sistem antrian yang terjadi di *buffer*. Hasil yang diharapkan adalah mengetahui seberapa besar dampak penerapan algoritma AQM pada *multimedia streaming* terhadap QoS. Dampak tersebut dibandingkan dengan metode konvensional (algoritma Drop-Tail) sehingga dapat ditunjukkan karakteristik masingmasing algoritma.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam tugas akhir ini terdiri dari beberapa tahap sebagai berikut.

Studi literatur

Kegiatan pengumpulan dan pengkajian terhadap referensi terkait topik tugas akhir yang diusulkan. Referensi tersebut berasal dari sumber ilmiah terpercaya, seperti buku materi, jurnal ilmiah, artikel ilmiah, pendapat para ahli, dan hasil penelitian terkait.

2. Identifikasi sistem

Identifikasi sistem disini berfungsi agar mengetahui berbagai macam faktor yang dapat mempengaruhi jaringan, sistem antrian pada *buffer*, penggunaan algoritma AQM pada sistem antrian di *buffer*, serta penggunaan algoritma Drop-Tail pada sistem antrian di *buffer*.

3. Pemodelan sistem

Sistem antrian yang dimodelkan merupakan bagian dari sistem multimedia. Sistem antrian ini dimodelkan menjadi model matematis agar dapat diselesaikan secara pendekatan *operation research*. Pemodelan sistem ini dirancang dengan bantuan referensi terkait. Pemodelan sistem ini berfungsi pula sebagai ilustrasi dari antrian virtual yang berada dalam *buffer*.

4. Pengujian sistem

Pengujian sistem bertujuan untuk menguji sistem yang telah dimodelkan dalam simulasi SimuLink. Setelah melakukan pengujian maka kemudian melakukan analisis dan evaluasi terhadap sistem antrian pada buffer dengan membandingkan algoritma AQM dengan algoritma konvensional atau algoritma yang umum digunakan (Drop-Tail). Setelah dilakukan analisis dan ditemukan kekurangan atau hasil yang kurang memuaskan atau bahkan kecacatan maka model sistem direvisi kembali mulai dari tahap identifikasi.

5. Penarikan kesimpulan dan saran

Penarikan kesimpulan mengacu pada data pengujian, analisis data, dan referensi terkait. Kesimpulan menunjukkan hasil kerja secara garis besar sesuai rumusan masalah yang telah dibuat. Selanjutnya, penarikan saran juga perlu dilakukan sebagai bentuk koreksi terhadap penelitian yang telah dilakukan dan pengembangan penelitian selanjutnya terkait topik serupa.

6. Penyusunan buku tugas akhir

Tahap ini merupakan tahap akhir dari serangkaian pelaksanaan tugas akhir. Penyusunan buku tugas akhir dilakukan sebagai bentuk laporan tertulis dari proses dan hasil kerja terkait topik yang diusulkan.

1.6 Sistematika Pembahasan

Pembahasan dalam tugas akhir ini dibagi dalam enam bab dengan sistematika sebagai berikut.

Bab I Pendahuluan

Pada bab ini dijelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Penunjang

Pada bab ini berisi tentang dasar teori yang merupakan penjelasan terkait *multimedia streaming*, *quality-of-service*, algoritma AQM, dan algoritma Drop-Tail, serta sistem antrian secara umum.

Bab III Perancangan Sistem

Pada bab ini membahas tentang sistem antrian pada buffer, model matematik dari sistem yang dirancang, dan simulasi sistem antrian dengan algoritma AQM.

Bab IV Simulasi Sistem

Bab ini berisi simulasi dari model sistem antrian yang telah dirancang dan memaparkan parameter, variabel, dan batasan atas model sistem antrian yang telah dirancang.

Bab V Pengujian Sistem

Pada bab ini berisi pengujian simulasi sistem antrian dengan algoritma AQM maupun algoritma Drop-Tail. Kemudian hasil simulasi tersebut dianalisis dan dibandingkan dengan mempertimbangkan QoS dari masing-masing sistem.

Bab VI Penutup

Pada bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh dari bab-bab sebelumnya. Saran diperlukan sebagai bentuk koreksi terhadap penelitian yang telah dilakukan dan pengembangan penelitian selanjutnya

1.7 Relevansi

Hasil dari yang didapat pada tugas akhir ini diharapkan dapat diimplementasikan pada *server* ITS dan solusi untuk meningkatkan QoS pada kasus *multimedia streaming* di kampus Institut Teknologi Sepuluh Nopember.

BAB 2 TEORI PENUNJANG

Sebuah penilitian memperlukan suatu teori yang mampu menunjang kelancaran pelaksanaan dari penelitian tersebut. Oleh karena itu, bab ini menjelaskan semua teori penunjang yang dijadikan acuan dari pelaksanaan penelitian tugas akhir ini.

2.1 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 berupa perhitungan yang menunjukkan performa dari sistem antrian tersebut.

2.1.1 Tujuan Sistem Antrian

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.

2.1.2 Notasi Kendall

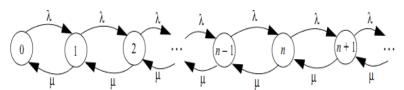
Untuk mengidentifikasi jenis sistem antrian yang dimodelkan biasanya dinyatakan dalam notasi Kendall. Kendall secara khusus membedakan sistem antrian berdasarkan notasi A/B/n/p/k. Kode huruf A mewakili jenis proses kedatangan. Kode huruf B mewakili jenis proses pelayanan. n menyatakan jumlah *server*. k menyatakan kapasitas sistem dan p menyatakan jenis populasi pelanggan (*finite/infinite*) [2]. Kode huruf yang biasa dipergunakan untuk mengisi A dan B antara lain.

- M = poisson (M merupakan singkatan dari "memoryless")
- D = deterministik (Kedatangan dan waktu pelayanan sudah ditentukan dengan pasti)
- Ek = Distribusi Erlang orde-k
- Hk = Distribusi hiper-eksponensial orde-k
- G = Distribusi 'umum'

Disiplin antrian merupakan aturan bagaimana *user* dilayani oleh *server*. Beberapa istilah digunakan mewakili beberapa disiplin antrian yang umum. FIFO mewakili urutan antrian 'first-in, first-out'. Pada antrian FIFO *user* yang dilayani sesuai urutan kedatangan. LIFO mewakili urutan antrian 'last in, first out'. Pada antrian LIFO *user* yang datang terakhir akan dilayani pertama. SIRO mewakili urutan antrian 'service in random order'.

2.1.3 Model Sistem Antrian Markovian

Model sistem antrian Markovian adalah sistem antrian dimana kejadian sistem berikutnya dipengaruhi hanya oleh kejadian sistem saat ini. Contoh kasus yang dapat dipresentasikan dengan Markov-chain adalah proses kelahiran dan kematian [2]. Dimana "populasi" memiliki nilai $\{0,1,2,3,\ldots\}$ pada sistem dan dinotasikan sebagai n. Transisi antar kejadian terjadi apabila unit populasi ini mengalami kenaikan atau penurunan nilai. Maksudnya adalah, ketika $n \geq 0$, maka kedatangan berikutnya (atau "kelahiran") adalah variabel acak yang terdistribusi eksponensial dengan laju λ_n . Pada saat terjadi kedatangan, maka nilai n berpindah menjadi n + 1. Ketika n ≥ 1 , maka keberangkatan berikutnya (atau "kematian") adalah variabel acak yang terdistribusi eksponensial dengan laju keberangkatan μ_n . Pada saat terjadi keberangkatan, maka nilai n berpindah menjadi n - 1. Kejadian-kejadian ini akan terus berulang secara kontinyu, dimana ilustrasi proses kelahiran dan kematian dapat dilihat pada Gambar 2.1.

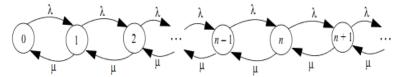


Gambar 2.1. Diagram transisi proses kelahiran dan kematian

Pada teori antrian, notasi n dapat diartikan sebagai jumlah pelanggan, notasi λ_n dapat diartikan sebagai laju kedatangan, dan μ_n dapat diartikan sebagai laju pelayanan. Berikut adalah model sistem antrian Markovian yang sederhana.

1) Single server (M/M/1)

Pada model antrian *single server*, diasumsikan bahwa waktu kedatangan antar pelanggan dan waktu pelayanan di *server* terdistribusi eksponensial.

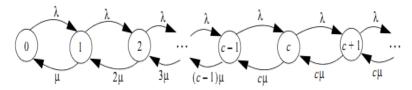


Gambar 2.2. Diagram transisi antrian *single server*

Seperti yang tampak dari Gambar 2.2, sistem antrian *single server* ini jika diibaratkan dengan proses kelahiran dan kematian maka nilai λ_n untuk $n = \{0, 1, 2, 3, \ldots\}$ adalah λ dan nilai μ_n untuk $n = \{1, 2, 3, \ldots\}$ adalah μ .

2) Multi server (M/M/c)

Pada model sistem antrian *multi server*, diasumsikan bahwa kedatangan adalah Poisson dengan laju λ , dengan sebanyak c *server*, dan setiap *server* independen dan identik dengan lama waktu pelayanan terdistribusi eksponensial dengan rata-rata $1/\mu$.

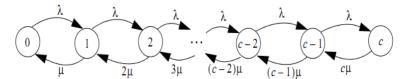


Gambar 2.3. Diagram transisi antrian multi server

Seperti yang tampak dari Gambar 2.3, sistem antrian *multi server* ini jika diibaratkan dengan proses kelahiran dan kematian maka nilai λ_n untuk $n = \{0, 1, 2, 3, ...\}$ adalah λ dan nilai μ_n untuk $n = \{1, 2, 3, ..., c\}$ adalah $n\mu$, serta μ_n untuk $n = \{c+1, c+2, ...\}$ adalah $c\mu$.

3) Kapasitas terbatas (M/M/c/K)

Pendekatan pada model antrian kapasitas terbatas ini mirip dengan pendekatan model antrian *multi server*, kecuali laju kedatangan pada λ_n bernilai 0 ketika $n \ge K$.



Gambar 2.4. Diagram transisi antrian multi server kapasitas terbatas

Pada Gambar 2.4, notasi c pada lingkaran menunjukkan jumlah *server* sedangkan notasi c di bawah lingkaran menunjukkan nilai K atau jumlah batasan kapasitas.

2.1.4 Little's Formula

$$L = \lambda W \tag{1}$$

Serta,
$$L_q = \lambda W_q \tag{2}$$

Oleh karena itu, penting didapatkan nilai dari salah satu dari empat parameter yang merepresentasikan performa sistem antrian memper-gunakan Little's Formula dan fakta bahwa $E[T] = E[T_q] + E[S]$.

2.1.5 Performa Sistem Antrian

Performa sistem antrian secara umum dapat dilihat dari nilai rata-rata waktu tunggu dalam sistem, rata-rata waktu tunggu dalam

antrian, rata-rata jumlah pelanggan dalam sistem, rata-rata jumlah pelanggan dalam antrian, proporsi jumlah pelanggan yang ditolak sistem, dan utilisasi *server*.

1) Utilisasi server (ρ)

Utilisasi *server* adalah probabilitas pelanggan tidak dilayani [3]. Atau dapat juga dikatakan sebagai proporsi *server* sibuk.

• Single server

$$\rho = \frac{\lambda}{\mu} \tag{3}$$

Atau,

$$\rho = 1 - P_0 \tag{4}$$

Keterangan:

 ρ = Utilisasi *server*

 $\lambda = \text{Laju kedatangan (/s)}$

 $\mu = \text{Laju pelayanan (/s)}$

 P_0 = Probabilitas tidak ada pelanggan

• Multi server

$$\rho = \frac{\lambda}{c\mu} \tag{5}$$

Keterangan:

 ρ = Utilisasi *server*

 $\lambda = \text{Laju kedatangan (/s)}$

 $\mu = \text{Laju pelayanan (/s)}$

c = Jumlah server

Kapasitas terbatas

$$\rho = \frac{\lambda}{c\mu} \tag{6}$$

Keterangan:

 ρ = Utilisasi *server*

 $\lambda = \text{Laju kedatangan (/s)}$

 μ = Laju pelayanan (/s)

c = Jumlah server

- Rata-rata jumlah pelanggan dalam sistem (L) dan antrian (Lq)
 Performa sistem antrian yang membahas tentang jumlah pelanggan
 dalam sistem maupun dalam antrian dapat dihitung dengan rumus
 berikut ini [3].
 - Single server

$$L = E[N] = \sum_{n=0}^{\infty} np_n$$

$$= (1 - \rho) \sum_{n=0}^{\infty} n\rho^n$$

$$\sum_{n=0}^{\infty} n\rho^n = \rho + 2\rho^2 + 3\rho^3 + \cdots$$

$$= \rho (1 + 2\rho + 3\rho^2 + \cdots)$$

$$= \rho \sum_{n=1}^{\infty} n\rho^{n-1}$$
(8)

Diketahui pada Persamaan (8) bahwa $\sum_{n=1}^{\infty} n\rho^{n-1}$ adalah turunan dari $\sum_{n=0}^{\infty} n\rho^n$ dengan hasil ρ . Oleh karena itu dapat dirumuskan bahwa,

$$\sum_{n=0}^{\infty} n\rho^n = \frac{1}{1-\rho} \tag{9}$$

sehingga,

$$\sum_{n=1}^{\infty} n\rho^{n-1} = \frac{1}{(1-\rho)^2} \tag{10}$$

sehingga,

$$L = \frac{\rho(1-\rho)}{(1-\rho)^2}$$

$$= \frac{\rho}{(1-\rho)} = \frac{\lambda}{\nu-\lambda}$$
(11)

Dengan menggunakan *Little's Formula* maka didapatkan, $L_q=\sum_{n=1}^\infty (n-1)p_n=\sum_{n=1}^\infty np_n-\sum_{n=1}^\infty p_n$

$$= L - (1 - p_0) = \frac{\rho^2}{1 - \rho} = \frac{\lambda^2}{\mu(\mu - \lambda)}$$
 (12)

Keterangan:

L = Jumlah pelanggan dalam sistem

Lq = Jumlah pelanggan dalam antrian

 λ = Laju kedatangan (/s)

 μ = Laju pelayanan (/s)

N = Jumlah pelanggan dalam sistem

 $P_{\rm N}$ = Probabilitas N pelanggan dalam sistem

 ρ = Utilisasi *server*

r =Beban kerja antrian

Multi server

Pada antrian *multi server* diketahui bahwa $r = \lambda/\mu$ atau $r = \rho/c = \lambda/c\mu$, dimana r adalah beban kerja antrian. Oleh karena itu rumus untuk mencari rata-rata panjang antrian di dalam sistem maupun antrian dengan Persamaan (12) didapatkan sebagai berikut.

$$L_{q} = \sum_{n=c+1}^{\infty} (n-c)p_{n}$$

$$= \sum_{n=c+1}^{\infty} (n-c) \frac{r^{n}}{c^{n-c}c!} p_{0}$$
(13)

$$L_q = \left(\frac{r^c \rho}{c!(1-\rho)^2}\right) p_0 \tag{14}$$

Untuk mendapatkan rumus L, maka digunakan *Little's Formula*

$$L = r + \left(\frac{r^c \rho}{c!(1-\rho)^2}\right) p_0 \tag{15}$$

Keterangan:

L = Jumlah pelanggan dalam sistem

 L_q = Jumlah pelanggan dalam antrian

r = Beban kerja antrian

c = Jumlah server

 ρ = Utilisasi *server*

 p_0 = Probabilitas tidak ada pelanggan

Kapasitas terbatas

Berdasarkan Persamaan (14) pada rumus antrian *multi* server kapasitas tak berhingga, didapatkan rumus sebagai berikut.

$$Lq = \sum_{n=c+1}^{K} (n-c)p_n$$

$$= \sum_{n=c+1}^{K} (n-c) \frac{\lambda^n}{c^{n-c}c!\mu^n} p_0$$
(16)

$$L_q = \frac{p_0 r^c \rho}{c! (1-\rho)^2} [1 - \rho^{K-c+1} - (1-\rho)(K-c+1)\rho^{K-c}]$$
 (17)

Dengan menggunakan *Little's Formula* dapat didapatkan nilai dari jumlah pelanggan dalam sistem. Namun, karena kapasitas antrian terbatas maka laju kedatangan bernilai 0 ketika kapasitas antrian telah dipenuhi. Oleh karena itu, perlu digunakan λ_{eff} untuk mencari nilai dari rata-rata jumlah pelanggan dalam sistem.

$$L = L_q + \frac{\lambda_{eff}}{\mu} = L_q + \frac{\lambda(1 - P_K)}{\mu} \tag{18}$$

Keterangan:

L = Jumlah pelanggan dalam sistem

Lq = Jumlah pelanggan dalam antrian

c = Jumlah server

 λ = Laju kedatangan (/s)

 μ = Laju pelayanan (/s)

 P_0 = Probabilitas tidak ada pelanggan

 ρ = Utilisasi *server*

K = Kapasitas server

r = Beban kerja antrian

- 3) Rata-rata lama waktu dalam sistem (W) dan antrian (Wq) Rata-rata lama waktu di dalam sistem maupun antrian dapat dicari menggunakan *Little's Formula* selama rata-rata jumlah pelanggan di dalam sistem atau di dalam antrian telah diketahui. Berikut rumus yang didapatkan [3].
 - Single server $W = \frac{L}{\lambda} = \frac{\rho}{\lambda(1-\rho)} = \frac{1}{\mu-\lambda}$ (19)

$$W_q = \frac{L_q}{\lambda} = \frac{\rho}{\mu(1-\rho)} = \frac{\rho}{(\mu-\lambda)} \tag{20}$$

Keterangan:

W = Lama waktu dalam sistem (s)

W_q = Lama waktu dalam antrian (s)

 $\lambda = \text{Laju kedatangan (/s)}$

 μ = Laju pelayanan (/s)

 ρ = Utilisasi *server*

L = Jumlah pelanggan dalam sistem

L_q = Jumlah pelanggan dalam antrian

Multi server

$$W_q = \frac{L_q}{\lambda} = \left(\frac{r^c}{c!(c\mu)(1-\rho)^2}\right) p_0 \tag{21}$$

$$W = \frac{1}{\mu} + \left(\frac{r^c}{c!(c\mu)(1-\rho)^2}\right) p_0 \tag{22}$$

Keterangan:

W = Lama waktu dalam sistem (s)

 $W_q = Lama$ waktu dalam antrian (s)

 $L_q = Jumlah pelanggan dalam antrian$

L = Jumlah pelanggan dalam sistem

 $\lambda = \text{Laju kedatangan (/s)}$

 μ = Laju pelayanan (/s)

r = Beban kerja antrian

c = Jumlah server

P₀ = Probabilitas tidak ada pelanggan

 ρ = Utilisasi *server*

Kapasitas terbatas

$$W = \frac{L}{\lambda_{eff}} = \frac{L}{\lambda(1 - P_K)} \tag{23}$$

$$W_q = W - \frac{1}{\mu} = \frac{L_q}{\lambda_{eff}} \tag{24}$$

Keterangan:

W = Lama waktu dalam sistem (s)

W_q = Lama waktu dalam antrian (s)

L = Jumlah pelanggan dalam sistem

 $L_{\text{q}} = Jumlah \ pelanggan \ dalam \ antrian$

 λ_{eff} = Laju kedatangan efektif (/s)

 λ = Laju kedatangan (/s)

 $\mu = \text{Laju pelayanan (/s)}$

P_K = Proporsi jumlah pelanggan yang ditolak

- 4) Proporsi jumlah pelanggan yang ditolak sistem (P_K) Nilai dari proporsi ini menunjukkan berapa probabilitas pelanggan yang tidak dapat dilayani oleh sistem serta tidak dapat masuk ke dalam antrian [3].
 - Kapasitas terbatas

$$P_K = \frac{\lambda^K}{c!c^{(K-c)}\mu^K} P_0 \tag{25}$$

Keterangan:

 P_K = Proporsi pelanggan yang ditolak sistem

K = Kapasitas antrian

c = Jumlah server

 P_0 = Probabilitas tidak ada pelanggan

 λ = Laju kedatangan (/s)

 μ = Laju pelayanan (/s)

5) Menentukan jumlah server (c)

Jumlah *server* menentukan performa dari sistem antrian. Semakin banyak jumlah *server* semakin baik pula performa sistem antrian [3]. Menentukan jumlah *server* dapat didapatkan dengan persamaan berikut.

$$c = r + \Delta$$
 ; $\Delta > 0$ (26)

Keterangan:

c = Jumlah server

r = Beban kerja antrian

 Δ = Jumlah *server* tambahan pada kondisi beban kerja berlebih (dapat berupa bilangan desimal untuk membulatkan jumlah *server*)

2.2 Multimedia Streaming

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi, audio dan video dengan alat bantu dan koneksi sehingga pengguna dapat berinteraksi, berkarya dan berkomunikasi. Seperti yang diilustrasikan oleh Gambar 2.5. *Streaming* adalah teknik mentransfer data agar dapat diproses yang terjadi secara konstan dan berkelanjutan. Oleh karena itu, *multimedia streaming*

adalah penggunaan teknik mentransfer data yang dapat berupa teks, suara, gambar, animasi, audio, maupun video dengan alat bantu dan koneksi yang terjadi secara berkelanjutan sehingga *user* dapat berinteraksi, berkarya, dan berkomunikasi.

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.

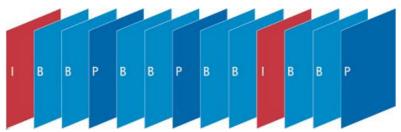


Gambar 2.5. Ilustrasi *multimedia streaming*

Video streaming dapat diartikan transmisi file video secara bekelanjutan yang memungkinkan video tersebut diputar tanpa menunggu file video tersebut tersampaikan secara keseluruhan. Secara sederhana, video streaming adalah layanan internet untuk menonton video tanpa mengunduh video tersebut untuk memutarnya. Akhir-akhir ini video streaming banyak diimplementasikan pada dunia pertelevisian untuk melakukan siaran dari website atau mengirimkan gambar siaran langsung melalui website atau disebut juga live streaming. Jadi gambar yang didapatkan dari siaran langsung, sesegera mungkin ditransmisikan dan dapat diputar melalui internet.

2.2.1 Metode Pengiriman Data pada Multimedia Streaming

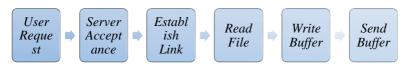
Metode Pengiriman Data Multimedia Streaming Secara Sederhana adalah dengan membagi video menjadi beberapa bagian paket yang di-*encode* sebelum dikirim dan kemudian paket tersebut di-*decode* agar dapat diputar [4]. Proses ini dilakukan berulang kali sampai paket video terkirim sepenuhnya.



Gambar 2.6. Ilustrasi pengiriman data video streaming

Pengguna internet yang menggunakan layanan *video streaming* menginginkan kualitas video yang baik dengan kecepatan yang memuaskan. Peningkatan kualitas ini yang menghasilkan berbagai penelitian dan penemuan terkait *video streaming* semakin berkembang, contohnya adalah format video H.264 [5]. H.264 adalah format video yang populer digunakan untuk *high definition digital video*. Codec yang berbasis format H.264 mengkompres video *file* sehingga hanya menggunakan separuh kapasitas MPEG-2 (standar DVD) untuk mengirim data dengan berkualitas yang sama.

2.2.2 Proses Multimedia Streaming di Buffer

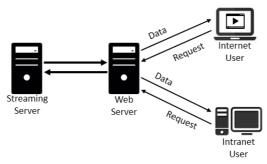


Gambar 2.7. Proses yang terjadi pada *video streaming*

Pada Gambar 2.7, dapat dilihat bahwa proses pengiriman paket data video *streaming* yang terjadi pada *buffer* melalui enam tahapan tersebut. Proses pengiriman paket diawali oleh pengguna internet yang

mengirimkan request. Kemudian jaringan akan terhubung setelah server telah mengakui dan menerima requests dari pengguna tersebut. Sebelum video dikirimkan akan melalui proses read file, yang berarti video akan dibaca sebelumnya oleh server sebelum dikirimkan melalui internet. Berikutnya, video tersebut akan dipecah menjadi beberapa bagian dan diencode sebelum dikirim menuju komputer pengguna internet melalui internet. Proses-proses tersebut terfokus pada buffer yang merupakan sutau memory pendukung.

Adapun yang penting diketahui dan diingat bahwa *buffer* adalah *memory* yang berfungsi sebagai media penyimpanan sementara antara dua *device* atau antara *device* dengan aplikasi [6]. Dengan kemampuan *buffer* yang demikian sehingga *buffer* mampu menjadi sistem antara natara paket-paket data yang berdatangan.



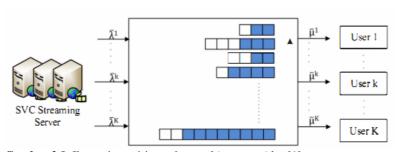
Gambar 2.8. Skema proses multimedia streaming

Skema yang ditampilkan pada Gambar 2.8 menunjukkan hubungan antara user dan server dalam jaringan komputer. Web server pada gambar tersebut berperan sebagai front-end server, sedangkan media (streaming) server berperan sebagai back-end server. Umumnya, User mengakses interface yang berada pada front-end server dan interface inilah yang men-trigger proses streaming pertama. Komunikasi antara front-end server dengan back-end server adalah dalam wujud request dan pengiriman paket data. Request dikirimkan dari front-end server menuju back-end server, sebaliknya paket data dikirimkan dari back-end server menuju front-end server. Media server dapat berperan ganda sebagai streaming server dan juga database server. Oleh karena itu, front-end server cukup mengirimkan permintaan ke back-end server untuk memproses request dari user.

2.3 Algoritma Active Queue Management (AQM)

Pendekatan Active Oueue Management (AOM) secara umum dibagi menjadi dua kategori, yaitu AOM berbasis antrian dan AOM berbasis beban [7]. 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. Dalam hal ini, beban dapat didefinisikan menjadi banyak hal, seperti perbedaan antara laju kedatangan dan laju pelayanan AQM antrian. berbasis antrian tergolong pada sulit mengkonfigurasikan 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, AOM berbasis beban dapat mempercepat respon, namun dapat menimbulkan ketidakstabilan sistem.

Algoritma AQM memiliki keunggulan dalam memproses pengiriman data *multi stream* video. Dengan kata lain, semakin banyak jumlah pengguna layanan video *streaming* maka semakin menonjol pula algoritma AQM.



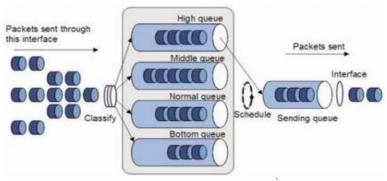
Gambar 2.9. Ilustrasi pengiriman data multi stream video [1]

Berdasarkan Gambar 2.8, *buffer* bekerja dengan menciptakan antrian virtual yang digunakan sebagai media penyimpanan sementara, sekaligus Gambar 2.8 mempermudah konversi prinsip kerja *buffer* menjadi model matematis dengan ilustrasi tersebut.

2.4 Algoritma Drop-Tail Queue Management

Drop-Tail Queue Management adalah algoritma manajemen antrian sederhana yang digunakan pada router, yang mana meniadakan paket berdasarkan pada panjang antrian. Pada penggunaan algoritma Drop-Tail, semua paket dianggap identik, dalam artian tidak ada prioritas pengiriman paket data [8]. 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 [8]. *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 untun permasalahan antrian penuh adalah dengan *router* meniadakan paket sebelum kapasitas antrian terpenuhi seluruhnya.



Gambar 2.10. Ilustrasi pengimplementasian algortima Drop-Tail [8]

2.5 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. QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu layanan [9].

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.

2.5.1 *Jitter*

Jitter merupakan variasi delay antar paket yang terjadi pada jaringan berbasis IP [10]. Besarnya nilai jitter sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (congestion) yang ada dalam jaringan tersebut. Semakin besar beban trafik di dalam jaringan menyebabkan semakin besar pula peluang terjadinya congestion, dengan demikian nilai jitter-nya menjadi semakin besar. Semakin besar nilai jitter menyebabkan nilai QoS semakin turun. Kategori kinerja jaringan berbasis IP dalam jitter versi Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) mengelompokkan menjadi empat kategori penurunan kinerja jaringan berdasarkan nilai jitter seperti terlihat pada Tabel 2.1 di bawah.

Tabel 2.1. Tabel performa jaringan berdasarkan *jitter* [10]

	-
Kategori <i>Jitter</i>	Peak Jitter
Sangat Bagus	0 ms
Bagus	75 ms
Sedang	125 ms
Jelek	225 ms

2.5.2 Bandwidth

Bandwidth merupakan kapasitas atau daya tampung ethernet agar dapat dilewati trafik paket data dalam jumlah tertentu [11]. Bandwidth juga bisa berarti jumlah konsumsi paket data per satuan waktu, yang dinyatakan dengan bit-per-second (bps), antara server dan client dalam waktu tertentu.

2.5.3 Latency

Latency dalam jaringan adalah hasil dari dua arah atau roundtrip (bolak-balik) [11]. Latency dapat diilustrasikan ketika komputer A terhubung ke komputer B, komputer A mengirimkan data ke komputer B, kemudian komputer B 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 2.2. Tabel performa jaringan berdasarkan *delay*

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

2.5.4 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. Oleh karena itu 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 [10].

Tabel 2.3. Tabel performa jaringan berdasarkan *packet-loss* [10]

Kategori Degradasi	Packet-loss
Sangat bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

2.5.5 Throughput

Throughput adalah bandwidth aktual yang terukur pada suatu ukuran waktu tertentu dalam satu hari menggunakan rute internet yang spesifik ketika sedang men-download suatu file.

BAB 3 PERANCANGAN SISTEM

3.1 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*. Skema dari *plant* yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.7 dan Gambar 2.8.

Sistem yang dirancang merupakan sistem antrian yang diadopsi dari jaringan komputer yang menyediakan layanan *video streaming*. Dengan batasan masalah yang telah dibahas pada Bab 1, maka sistem antrian yang dirancang merupakan sistem antrian pada buffer dimana delay dan sistem antrian lain yang menunjang sistem diabaikan.

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.

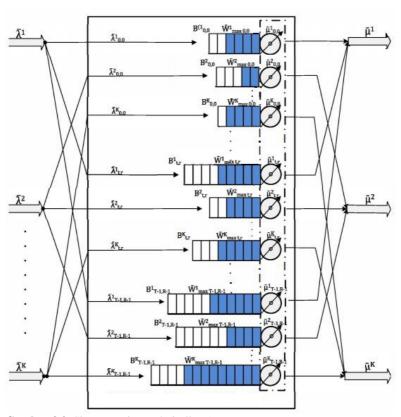


Gambar 3.1. Proses *streaming* yang dimodelkan dan disimulasikan

Berdasarkan Gambar 3.1 di atas, 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 disebut sebagai paket data dan *multi stream* yang terjadi pada *buffer* dimodelkan menjadi sistem antrian dengan n *user* dan c *server*.

Kepadatan pada proses *video streaming* terjadi pada proses *write buffer* dan dapat juga terjadi pada proses *send buffer* ketika jumlah paket

data yang diproses oleh *buffer* melebihi kapasitas fisik *buffer*. Permasalahan kepadatan ini apabila dikonversi menjadi permasalahan pada kasus sistem antrian menjadi suatu kasus saat jumlah pelanggan yang datang tidak semua dapat dilayani oleh *server* dan tidak semua dapat menunggu di dalam antrian karena keterbatasan kapasitas antrian.



Gambar 3.2. Sistem antrian pada *buffer* [1]

Oleh karena itu, pada tugas akhir ini penulis melakukan penelitian terkait permasalahan kepadatan tersebut. Permasalahan kepadatan pada *multimedia streaming* dimodelkan sebagai sistem antrian dan menggunakan algoritma AQM berbasis antrian sebagai pemecah permasalahan. Pada tahap pengujian, model sistem antrian yang

menggunakan algoritma AQM berbasis antrian dibandingkan dengan model sistem antrian yang menggunakan algoritma Drop-Tail sebelum menarik kesimpulan.

3.2 Perancangan Model Matematis Sistem

Model matematis sistem berfungsi untuk menginterpretasikan sistem dan permasalahannya dalam bentuk numerik yang lebih detil. 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. Notasi Kendall ini didapat berdasarkan ilustrasi pada Gambar 3.2 yang menunjukkan karakteristik sistem antrian.

3.2.1 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* yang telah dirumuskan di Bab 2 dan dikutip dari Persamaan (24) 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.

$$\min \sum_{n=1}^{c} \overline{W}_{q_n} + \frac{1}{\overline{\mu}_n}$$
 (27)

Keterangan:

 \overline{W}_q = Rata-rata waktu paket data di dalam antrian

 $\bar{\mu}$ = Rata-rata laju pelayanan server

c = Jumlah *server*

Persamaan (27) 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*. Oleh karena itu performa sistem antrian menjadi lebih baik.

3.2.2 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*.

$$\overline{W}_n \le \overline{W}_{max} \tag{28}$$

$$\frac{1}{\overline{\mu}_{n}} \le \frac{1}{\overline{\mu}_{max}} \tag{29}$$

$$\sum_{n=1}^{c} K_n \le K_{max} \tag{30}$$

Keterangan:

 \overline{W} = Rata-rata waktu paket data di dalam sistem

 \overline{W}_{max} = Rata-rata waktu *time-out* paket data di dalam sistem

 $\bar{\mu}$ = Rata-rata laju pelayanan server

 $\bar{\mu}_{max} = \text{Rata-rata laju paket yang } time-out \text{ sebelum atau pada proses pelayanan } server$

K = Kapasitas antrian pada masing-masing server

 K_{max} = Kapasitas fisik *buffer*

c = Jumlah server

Persamaan (28) digunakan untuk memastikan waktu paket data dalam sistem yang melalui antrian virtual tidak melebihi waktu *time-out* yang 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 (dilewati) atau dilakukan request ulang terkait paket data tersebut.

Persamaan (29) digunakan untuk memastikan bahwa waktu pelayanan tidak melebihi waktu *time-out* yang diatur untuk mengkontrol 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 semakin singkat pula delay dari sistem antrian. Namun, hal ini berimbas pada packet loss yang memperburuk kualitas video. Sebaliknya, semakin lama waktu time-out maka semakin baik kualitas video, akan tetapi delay menjadi semakin tinggi.

Sedangkan Persamaan (30) mempertegas eksistensi *buffer*. Diketahui bahwa 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.

3.3 Perancangan Simulasi

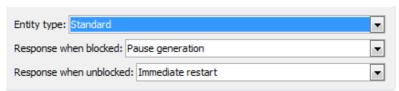
Dalam merancang simulasi sistem antrian yang didasari oleh model matematis pada Sub-Bab 3.2 dan mempertimbangkan sistem antrian. Oleh karena itu harus didapatkan data awal sebagai input dari simulasi sistem tersebut. Data-data yang penting didapat adalah rata-rata jumlah *request* dari para *user* yang 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 (Persamaan (26)).

3.3.1 Waktu Antar Kedatangan

Berdasarkan *incoming request* dari satu atau lebih user yang diolah maka dapat diketahui besarnya waktu antar kedatangan. Hasil dari data yang telah diolah ini kemudian dimasukkan ke dalam *block parameters* bernama *time-based entity generator* pada SimuLink.



Gambar 3.3. Gambar block parameters time-based entity generator



Gambar 3.4. Gambar kotak dialog untuk memasukkan tipe entity

Gambar 3.4 menunjukkan tipe *entity* dan aksi yang diambil ketika terjadi kepadatan, yang digunakan dalam simulasi sistem antrian ini.

Generate entities with: Intergeneration time from dialog				
Distribution: Exponential Initial seed: 91				
Mean: 81				
Generation event priority: 301				
Generate entity at simulation start				

Gambar 3.5. Gambar kotak dialog untuk memasukkan nilai input

Gambar 3.5 menunjukkan kotak dialog untuk memasukkan hasil dari data yang telah diolah menjadi waktu antar kedatangan paket data.

3.3.2 Kapasitas Antrian

Rata-rata nilai *memory temporary buffer* pada tahap pengambilan data yang kemudian diolah dan dikonversi maka didapatkan nilai kapasitas antrian. Kapasitas antrian bernilai identik pada setiap antrian virtual sesuai pada Sub-Bab 1.3.



Gambar 3.6. Gambar block parameters FIFO queue

FIFO Queue	Timeout	Statistics
Capacity:		
50		

Gambar 3.7. Gambar kotak dialog untuk kapasitas maksimal

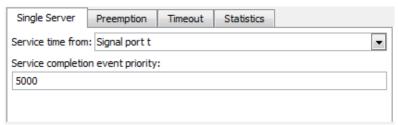
Data yang dimasukkan dalam kotak dialog ini telah dikonversi ke dalam satuan unit atau lebih dikenal dengan pelanggan. Seperti yang telah tertera pada Sub-Bab 3.1 bahwa satuan unit berupa paket data. Ukuran dan jumlah data pada paket data ditentukan pada bab berikutnya.

3.3.3 Waktu Pelayanan

Idealnya waktu pelayanan diambil dari nilai *delay* pada saat pemrosesan request. Data *delay* tersebut kemudian diolah dan dijadikan nilai input dari waktu pelayanan simulasi sistem antrian.



Gambar 3.8. Gambar block parameters single server



Gambar 3.9. Gambar kotak dialog untuk memasukkan nilai input

Nilai input pada simulasi ini bergantung nilai acak yang terdistribusi secara eksponensial dengan nilai *mean* yang identik untuk masing-masing antrian virtual.

3.3.4 Waktu *Time-out* Paket Data di dalam Sistem

Nilai ini didapatkan berdasarkan *delay* terbesar yang ditoleransi oleh sistem ketika proses pengambilan data. Data tersebut dijadikan input simulasi setelah dikonversi berdasarkan *simulation time* pada SimuLink.



Gambar 3.10. Gambar block parameters setting time-out

Schedule Timeout	Statistics	
Timeout tag:		
TO1		
Timeout interval from	: Dialog	•
Timeout interval:		
3630		
Timeout event priority	y:	
7000		
If timeout is already s	cheduled: Warn and reschedule	•
If entity has no destir	nation when timeout occurs: Warn and discard the entity	•

Gambar 3.11. Gambar kotak dialog untuk delay maksimal

Berdasarkan Gambar 3.11, selain memasukkan nilai *delay* terbesar yang ditoleransi sistem, *block parameters* ini juga dapat mengatur aksi yang diambil ketika *entity* melebihi *delay* maksimal.

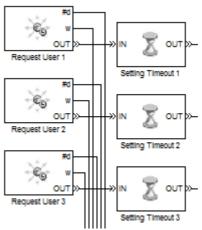
3.3.5 Jumlah Server

Jumlah server dapat ditentukan dengan menghitung beban kerja antrian dan menentukan bilangan yang membulatkan beban kerja tersebut (Persamaan (26)). Jumlah server menentukan kelancaran proses simulasi sistem antrian ini.

3.3.6 Proses Multimedia Streaming pada Simulasi

Dengan meninjau Gambar 2.7 dan Gambar 3.1, 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.

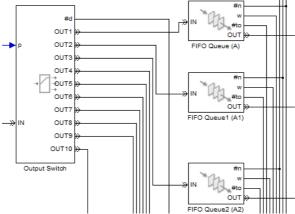
A. Read file



Gambar 3.12. Peran read file pada simulasi sistem

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

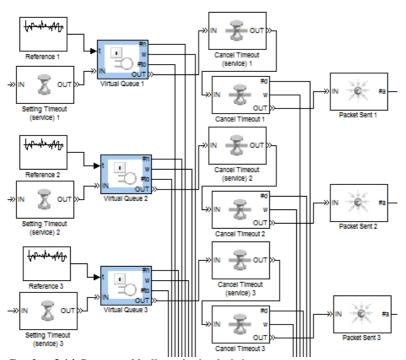
B. Write buffer



Gambar 3.13. 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 3.14. 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.

BAB 4 SIMULASI SISTEM

Bab ini membahas terkait simulasi sistem yang meliputi spesifikasi *server* dan pengolahan data, simulasi sistem antrian dengan penerapan algoritma AQM dan simulasi sistem antrian dengan penerapan algoritma Drop-Tail.

1.1 Spesifikasi Server dan Pengolahan Data

Seperti yang telah dijelaskan pada Bab 3 terkait perancangan sistem, data awal didapatkan dengan melihat performa dari sistem *video streaming* yang diambil sesuai kebutuhan perancangan sistem. Dalam sistem antrian, data awal yang dibutuhkan adalah laju kedatangan, laju pelayanan, kapasitas *server*, dan waktu *time-out. Video streaming server* yang digunakan untuk mengambil data adalah *server local host* yang dimiliki oleh laboratorium program studi Teknik Multimedia dan Jaringan (TMJ) yang bertempat di ruangan B-201.

4.1.1 Spesifikasi *Server* Kirby

Kirby adalah *server local host* yang memiliki layanan berupa website, *database server*, *multimedia streaming*, dan lain sebagainya. Alamat dari Kirby ini dapat diakses selama menggunakan internet dari kampus Institut Teknologi Sepuluh Nopember (ITS) dengan *log-in user* dan *password* dari civitas akademika, serta menggunakan *proxy* ITS. Alamat Kirby adalah http://10.122.1.99/. Berikut ini adalah spesifikasi *server* dari Kirby.

Tabel 4.1. Spesifikasi *server* Kirby

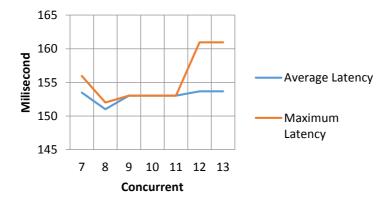
Processor	Intel Dual Core E6800 series
Memory	4 GB
Storage	HDD 150 GB
OS	Ubuntu 14.04.3 LTS
Ethernet	1 Gbps
Webserver	Nginx/1.4.6 Ubuntu

4.1.2 *Latency*

Latency digunakan sebagai rata-rata waktu pelayanan, karena jika seluruh proses yang berada di simulasi selain dari waktu pelayanan dalam antrian virtual dianggap 0, maka rata-rata nilai latency sama dengan rata-rata lama waktu pelayanan. Pada tahapan pengambilan data, didapatkan nilai latency adalah sebagai berikut.

Tabel 4.2. Tabel nilai rata-rata dan maksimal latency

Concurrent	Average Latency (ms)	Maximum Latency (ms)
7	153.49	155.96
8	151.01	152
9	153.03	153.03
10	153.03	153.03
11	153.03	153.03
12	153.67	160.96
13	153.67	160.96
Average	152.99	155.5671429



Gambar 4.1. Grafik rata-rata dan maksimal *latency*

Berdasarkan Tabel 4.2, nilai rata-rata *latency* adalah sebesar 152,99 milidetik. Perlu diingat bahwa *latency* adalah nilai nolak-balik (*round-trip*). Karena rata-rata waktu pelayanan merupakan waktu yang terjadi secara satu arah. Oleh karena itu, rata-rata waktu pelayanan dapat diasumsikan sebagai nilai rata-rata *latency* yang dibagi dua, yaitu kurang lebih 76 milidetik. Apabila dikonversi menjadi laju pelayanan maka laju pelayanan bernilai 13,16 paket data setiap detik.

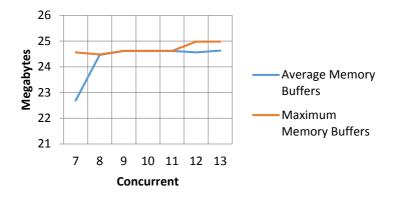
Adapun, nilai latency terbesar selama pengambilan data digunakan sebagai nilai maksimal *time-out*, yaitu 3,63 detik, karena selama pengambilan data tidak ada nilai lebih besar dari nilai tersebut. Oleh karena itu dapat dianggap jika nilai *latency* lebih besar maka tidak ada paket data yang terkirim. Nilai maksimal *latency* terbesar inilah yang dianggap sebagai *delay* terbesar yang dapat ditoleransi sistem.

4.1.3 Memory buffer

Memory buffer memiliki nilai sementara dalam artian bersifat dinamis atau berubah-ubah. Memory buffer berfungsi sebagai media penyimpanan sementara sebelum mengirimkan data kepada user dan dapat dianggap sebagai antrian pengiriman data menuju user. Seperti yang telah dijelaskan pada Sub-Bab 3 tentang perancangan simulasi, nilai dari memory buffer (temporary) tidak dapat lebih besar dari nilai maksimum memory buffer yang dialokasikan. Adapun hasil pengambilan data terkait memory buffer (temporary) adalah sebagai berikut ini.

Tabel 4.3. Tabel nilai rata-rata dan maksimal *memory buffer*

Concurrent	Average Memory Buffers (MB)	Maximum Memory Buffers (MB)
7	22.69	24.56
8	24.46	24.48
9	24.62	24.62
10	24.62	24.62
11	24.62	24.62
12	24.56	24.98
13	24.63	24.98
Average	24.31428571	24.69428571



Gambar 4.2. Grafik rata-rata dan maksimal *memory buffer*

Pada pengambilan data dari *server* Kirby, didapatkan nilai ratarata *memory buffer* (*temporary*) sebesar kurang lebih 24,3 Megabytes. Nilai ini yang nanti digunakan sebagai acuan dari model antrian pada simulasi yang dibuat, seperti alokasi kapasitas masing-masing antrian virtual.

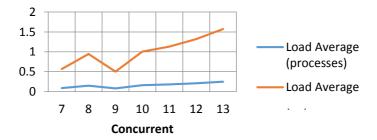
4.1.4 *Load*

Load atau beban kerja server dalam sistem ini digunakan sebagai pembobotan untuk setiap data. Beban kerja disini merupakan beban CPU memproses seluruh proses yang sedang dijalankan oleh server (komputer). Berikut ini adalah hasil pengambilan data terkait beban kerja.

Tabel 4.4. Tabel nilai rata-rata beban kerja *server*

Concurrent	Load Average (processes)	Load Average (KB)
7	0.09	0.566304
8	0.15	0.94384
9	0.08	0.503381333
10	0.16	1.006762667
11	0.18	1.132608
12	0.21	1.321376

Concurrent	Load Average (processes)	Load Average (KB)
13	0.25	1.573066667
Average	0.16	1.006762667



Gambar 4.3. Grafik nilai rata-rata beban kerja server

Berdasarkan Tabel 4.4 di atas dan konversi beban kerja *server* menjadi satuan byte, didapatkan rata-rata nilai beban kerja sebesar kurang lebih 1 KB. Oleh karena itu, diasumsikan bahwa setiap satu satuan data bernilai 1 KB. Nilai dari kapasitas antrian juga ditentukan dengan nilai dari beban kerja ini. Jika kapasitas antrian dikonversi berdasarkan nilai beban maka total kapasitas dari masing-masing antrian berjumlah 24000 antrian. Namun, nilai kapasitas antrian ini masih perlu dikonversi kembali bergantung pada berapa jumlah data pada setiap paket data dan jumlah data pada paket data akan dibahas pada Bab 5 tentang pengujian sistem.

4.1.4 Incoming User Requests

Waktu antar kedatangan dari paket data dapat didapatkan dengan mengambil dan mengolah data yang bersumber pada rata-rata waktu antar kedatangan *requests*. Data ini diambil pada proses pertama *multimedia streaming*, yaitu *user request* (Gambar 2.7). Berikut ini adalah data yang didapatkan terkait *incoming request*.

Tabel 4.5. Tabel nilai rata-rata waktu antar kedatangan request

Concurrent	Average Incoming Request	
7	63.77	
8	66.75	
9	66.75	
10	66.75	
11	63.08	
12	122.36	
13	123.21	
Average	81.81	

Berdasarkan Tabel 4.5, didapatkan rata-rata waktu antar kedatangan senilai kurang lebih 81 milidetik. Apabila dikonversi menjadi laju kedatangan maka laju kedatangan bernilai 12,35 paket data setiap detik.

4.2 Simulasi Sistem Antrian dengan Algoritma AQM

Pada Sub-Bab 3.3 tentang perancangan simulasi, telah dijelaskan peran proses *multimedia streaming*. Pada Sub-Bab ini, menjelaskan jumlah *server* yang digunakan pada simulasi, input masing-masing *block parameter* dan penjelasan terkait penggunaan algoritma AQM yang diterapkan di simulasi sistem antrian.

4.2.1 Keterangan Parameter Simulasi Sistem

Dalam melakukan penelitian ini jumlah antrian virtual yang dibuat berjumlah 9. Jumlah antrian virtual ini sengaja digunakan agar dapat meneliti kepadatan yang terjadi. Adapun, data yang ditampilkan pada tahap pengambilan dan pengolahan data sendiri merupakan data yang memiliki variasi nilai tidak jauh antara satu dengan yang lain.

Pada simulasi sistem antrian yang dibuat, waktu antar kedatangan yang dimasukkan pada *block parameter time-based entity generator* bernilai 81 dengan nilai awal berupa bilangan acak. Berdasarkan pada Sub-Bab 4.1.2, waktu pelayanan yang digunakan pada

simulasi sistem antrian bernilai 76 dan waktu time-out dalam simulasi sistem antrian bernilai 3630.

Kapasitas antrian pada simulasi sistem ini harus menyesuaikan paket data. Apabila paket data bernilai 60, maka kapasitas antrian pada antrian virtual bernilai 44 dan 45. Pada 5 antrian virtual memiliki kapasitas antrian sebanyak 44 dan 4 antrian virtual sisanya memiliki kapasitas antrian sebanyak 45. Oleh karena itu dapat disimpulkan bahwa kapasitas antrian menyesuaikan nilai paket data.

Tabel 4.6. Nilai parameter simulasi sistem antrian

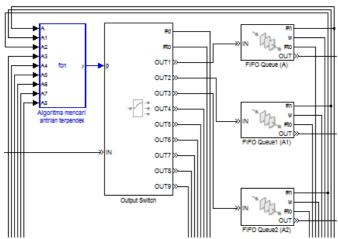
Notasi	Keterangan	Nilai
$^{1}/_{\lambda_{n}}$	Waktu antar kedatangan (ms)	81
$^{1}/\mu_{n}$	Waktu pelayanan antrian virtual (ms)	76
$^{1}/_{\mu_{max}}$	Waktu time-out paket data (ms)	3630
с	Jumlah server	9
K_n	Kapasitas antrian virtual	24300 9 x (paket data)

4.2.2 Penerapan Algoritma AQM

Algoritma AQM berbasis antrian yang diterapkan pada simulasi sistem antrian yang dibuat berada pada proses *write buffer* dan juga pada *send buffer*. Nilai-nilai pada *block parameter* simulasi pembanding ini bernilai identik dengan simulasi sistem antrian dengan algoritma AQM.

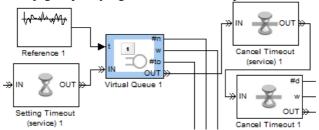
Pseudocode dalam mencari antrian terpendek

A(n) adalah nilai kapasitas antrian yang tersisa dan B(0) merepresentasikan peniadaan paket data. Oleh karena itu, pseudocode di atas menunjukkan bahwa paket data yang datang diarahkan untuk masuk pada antrian virtual dengan jumlah antrian terpendek.



Gambar 4.4. Penggunaan algoritma mencari antrian terpendek

Penggunaan algoritma pada Gambar 4.4 dan pseudocode di atas adalah untuk mencari antrian terpendek. Oleh karena itu, paket data yang datang dialokasikan pada rute yang menuju antrian virtual yang memiliki antrian terpendek atau terkosong. Penyebaran transmisi paket data ini dapat dilihat juga seperti yang telah diilustrasikan pada Gambar 3.2.



Gambar 4.5. Gambar pengaturan waktu time-out pada server

Sesuai dengan salah satu syarat yang telah dibahas pada Sub-Bab 3.2 terkait model matematis bahwa rata-rata waktu pelayanan dibatasi

untuk mempertahankan rata-rata nilai *jitter*. Oleh karena itu dalam simulasi sistem ini dibuat *time-out* kedua untuk membatasi hal tersebut. Oleh karena itu, dibuatlah *time-out* kedua dengan nilai pada Gambar 4.5.

Pseudocode untuk menentukan nilai time-out kedua

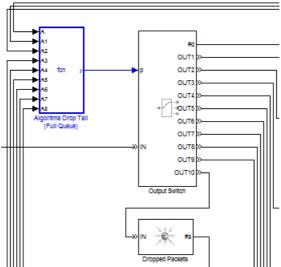
$$K = 100 * U / IN$$
 $T = 90 + (10 * K)$
if $IN == 0$
 $T = 90$
elseif $T >= 225$
 $T = 224$

U adalah nilai dari total paket data yang telah *time-out* dan IN adalah paket data yang telah atau sedang berada di dalam sistem. Konstanta (K), pada pseudocode, yang digunakan bernilai 90 dan kofaktor bernilai 10. Oleh karena itu, ketika tidak ada paket data yang hilang karena *time-out* maka nilai *time-out* untuk lama waktu pelayanan adalah 90 milidetik. Pseudocode di atas juga menunjukkan bahwa *time-out* kedua ini menjaga kualitas *jitter* di atas kategori jelek.

4.3 Simulasi Sistem Antrian dengan Algoritma Drop-Tail

Algoritma Drop-Tail yang diterapkan pada simulasi sistem antrian yang dibuat, sebagai algoritma pembanding, berada pada proses *write buffer*. Perlu diketahui bahwa parameter dari simulasi sistem antrian dengan penerapan algoritma Drop-Tail bernilai identik dengan simulasi sistem antrian dengan penerapan algoritma AQM.

Pseudocode untuk menghasilkan lock-out dan full queue



Gambar 4.6. Penerapan script untuk lock-out dan full queue

Gambar 4.6 menunjukkan karakteristik algoritma Drop-Tail, yaitu *lock-out* dan *full queue*. Pseudocode diatas menunjukkan bahwa jika antrian sudah penuh maka paket data yang datang akan ditiadakan tanpa mempedulikan waktu *time-out*. Paket data yang ditiadakan dapat diabaikan atau di-*request* ulang tergantung pada kebijakan yang dianut oleh *server*. Dimana pada simulasi sistem ini, kebijakan tersebut dianggap akan mengabaikan paket data yang ditiadakan.

BAB 5 PENGUJIAN SISTEM

Bab pengujian sistem dibagi menjadi 2 bagian. Pada bagian pertama dijelaskan mengenai tujuan pengujian dan pengujian apa yang dilakukan. Sedangkan bagian kedua membahas terkait analisis simulasi dan hubungannya terhadap QoS. Dari pengujian ini dapat ditunjukkan kelemahan, kelebihan, karakteristik, dan kualitas layanan masing-masing algoritma.

5.1 Tujuan Pengujian

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. Sedangkan untuk algoritma pembanding, yaitu algoritma Drop-Tail, dilakukan pengujian dengan parameter dan variabel yang sama. Hal ini ditujukan untuk mengetahui QoS dari sistem antrian pada kondisi yang hampir sama. Dari pengujian tersebut kemudian dapat dibandingkan dan dianalisis kekurangan, kelebihan, dan batasan-batasan dari masing-masing algoritma.

5.2 Pengujian QoS Sistem Antrian

Pengujian dilakukan dengan memberikan asumsi besarnya masing-masing paket data. Hal ini dilakukan untuk mendapatkan nilai kapasitas antrian (K), seperti pada Tabel 4.6. Pengujian juga dilakukan dengan menggunakan dua ukuran paket data yang berbeda untuk menunjukkan sensitifitas dari sistem antrian yang dirancang dalam simulasi, yang mana dibantu dengan tampilan grafik nantinya.

Diasumsikan nilai paket data yang berdatangan sebesar 25 KB dan 45 KB. Dengan meninjau persamaan yang tercantum pada Tabel 4.6, nilai total kapasitas antrian (K_{max}) masing-masing sebesar 972 dan 540. 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*.

5.2.1 Pengujian Simulasi Algoritma AQM

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

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 5.2. Hasil simulasi sistem antrian AOM (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	

Nilai dari *packet-loss*, total *delay*, dan *jitter* pada Tabel 5.1 dan Tabel 5.2 diambil dari nilai rata-rata hasil simulasi yang dilampirkan pada lampiran C.1. 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).

$$Packet - loss = \frac{drop + TO}{IN} x \, 100$$

Adapun total *delay* adalah lama waktu di dalam sistem. Oleh karena itu, nilai total *delay* dapat diselesaikan dengan *little's formula*. Sedangkan *jitter* merupakan variasi *delay* yang terjadi, dimana variasi ini

adalah *delay* yang terjadi antar penerimaan paket data. Oleh karena itu, dapat ditentukan bahwa nilai *jitter* sama dengan lama waktu pelayanan.

5.2.2 Pengujian Simulasi Algoritma Drop-Tail

Hasil pengujian yang dilakukan pada simulasi sistem antrian dengan menggunakan algoritma Drop-Tail adalah sebagai berikut.

Tabel 5.3. 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 5.4. Hasil simulasi sistem antrian Drop-Tail (K = 108)

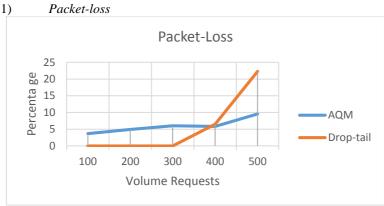
Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	0	340.494951	51.21627
200	0	451.646844	57.76633
300	0	667.709829	54.34265
400	5.02422201	823.726267	71.52556
500	41.1002635	2169.33842	68.2211

Sama halnya dengan bahasan pada Sub-Bab 5.2.1 bahwa nilai dari *packet-loss*, total *delay*, dan *jitter* pada Tabel 5.3 dan Tabel 5.4 diambil dari nilai rata-rata hasil simulasi yang dilampirkan pada lampiran C.2.

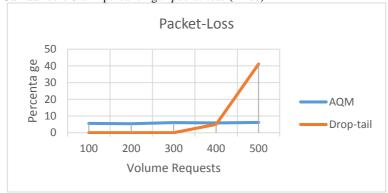
Dengan demikian, nilai QoS pada tabel-tabel di atas terhadap *volume requests* menunjukkan kualitas layanan dari model sistem antrian yang dirancang.

5.2.3 Analisis Perbandingan Algoritma pada Sistem

Pada Sub-Bab ini membahas terkait analisis dari hasil simulasi yang telah dibahas pada Sub-Bab 5.2.1, Sub-Bab 5.2.2, lampiran C.1, dan lampiran C.2. Analisis dibantu dengan bantuan grafik yang menunjukkan karakteristik masing-masing algoritma. Berikut ini adalah analisis dari hasil simulasi masing-masing algoritma berdasarkan QoS.



Gambar 5.1. Grafik perbandingan *packet-loss* (K=60)



Gambar 5.2. 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.

Dengan memperhatikan waktu simulasi dan jumlah paket yang telah dan sedang berada dalam sistem maka grafik-grafik pada Gambar 5.1 dan Gambar 5.2 dikarenakan pada beda nilai dari faktor-faktor tersebut.

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). Oleh karena itu 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. Oleh karena itu, ketika *volume requests* mencapai nilai 500 maka pada sistem antrian algoritma Drop-Tail baru terjadi penumpukan paket data yang hampir sama terjadi pada sistem antrian algoritma AQM.

Kategori sistem antrian dengan algoritma AQM berbasis antrian berdasarkan Tabel 2.1 tergolong sedang cenderung baik. Sedangkan kategori sistem antrian dengan algoritma Drop-Tail tergolong sedang cenderung buruk.

2) Total delay

Meninjau QoS total *delay* yang ditunjukkan pada Gambar 5.3 dan Gambar 5.4 di bawah.



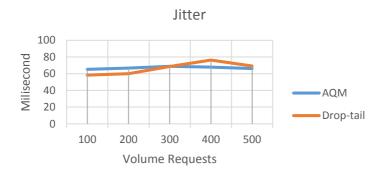
Gambar 5.3. Grafik perbandingan total *delay* (K=60)



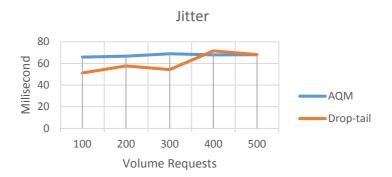
Gambar 5.4. Grafik perbandingan total *delay* (K=108)

Ditemui kondisi yang sama pada Gambar 5.1 dan Gambar 5.2 ditunjukkan pada Gambar 5.3 dan Gambar 5.4. 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. Dengan faktor yang sama dengan pada analisis QoS *packet-loss*, *trend* ini terjadi pula pada QoS total *delay*.

3) *Jitter*QoS *jitter* yang ditunjukkan pada Gambar 5.5 dan Gambar 5.6 di bawah.



Gambar 5.5. Grafik perbandingan *jitter* (K=60)



Gambar 5.6. Grafik perbandingan *jitter* (K=108)

Nilai *jitter* pada kedua sistem antrian dapat dianggap stabil. Penaikan dan penurunan nilai *jitter* tidak berubah secara signifikan. Oleh karena itu, dapat ditentukan bahwa kedua sistem antrian memiliki laju pengiriman data yang hampir sama. Dalam hal ini sistem antrian dengan algoritma Drop-Tail lebih baik dibandingkan sistem antrian dengan algoritma AQM berbasis antrian. Namun secara keseluruhan, kedua sistem antrian tergolong pada kategori baik.

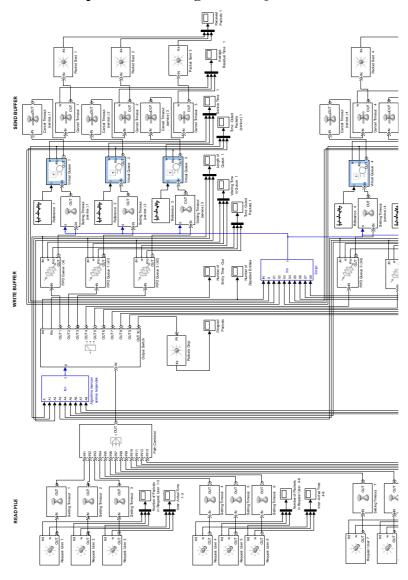
Berdasarkan analisis-analisis tersebut menunjukkan bahwa algoritma AQM dapat mengatur kepadatan atau *volume requests* lebih baik dibandingkan dengan Drop-Tail, yang ditunjukkan berdasarkan waktu simulasi terhadap *volume requests*. Oleh karena itu, dapat ditentukan apabila kondisi kedua sistem antrian adalah sama maka sistem antrian dengan algoritma AQM berbasis antrian lebih baik dibandingkan sistem antrian dengan algoritma Drop-Tail.

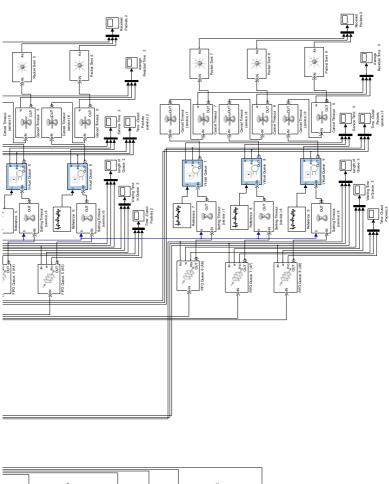
LAMPIRAN A

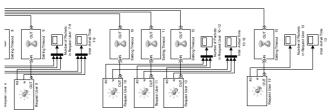
A.1 Script pada Simulasi Algoritma AQM

```
function y = fcn(A,A1,A2,A3,A4,A5,A6,A7,A8)
%fungsi y dengan variabel A0 s/d A7.
%Algoritma
           untuk mencari panjang antrian
terpendek.
N = [A A1 A2 A3 A4 A5 A6 A7 A8];
T = [60 \ 60 \ 60 \ 60 \ 60 \ 60 \ 60 \ 60];
R = T - N;
[B,C] = \max(R);
if max(B) \le 0
    y = 10;
else
    y = C;
end
%Syarat apabila antrian telah penuh maka paket
data yang masuk berikutnya akan di-drop
function T = fcn(IN, U, U1, U2, U3, U4, U5, U6, U7, U8)
%fungsi T dengan variabel U0 s/d U8 dan IN.
%Script
          untuk
                  menentukan
                                 nilai
                                          Time-out
berdasarkan persentase paket yang Time-out.
K = 100*(sum([U U1 U2 U3 U4 U5 U6 U7 U8]))/IN;
T = 90 + (10 * K);
if IN == 0
    T = 90;
elseif T >= 225
    T = 224;
end
%Untuk menghindari error 0/0 dan menjaga kualitas
jitter.
```

A.2. Tampilan Simulasi Algoritma AQM di SimuLink





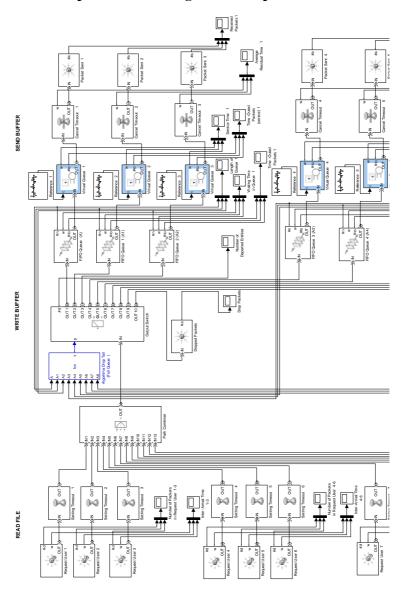


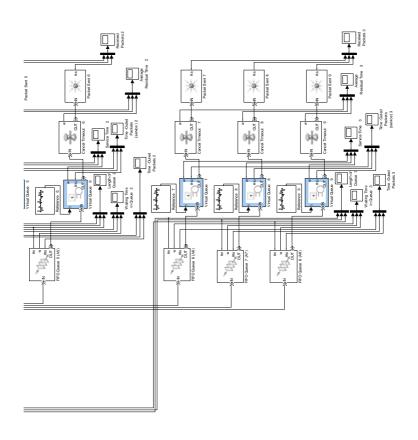
LAMPIRAN B

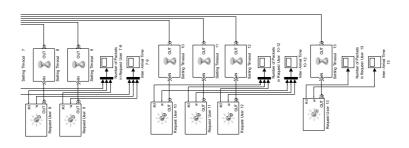
B.1 Script pada Simulasi Algoritma Drop-Tail

```
function y = fcn(A, A1, A2, A3, A4, A5, A6, A7,
A8)
%Fungsi y dengan variabel A0 s/d A7.
%Algoritma untuk menampilkan karakteristik Lock-
out dan Full queue.
N = [A A1 A2 A3 A4 A5 A6 A7 A8];
T = [60 \ 60 \ 60 \ 60 \ 60 \ 60 \ 60 \ 60];
R = T-N;
y=0;
for i=1:9
    if R(i) > 0
        v=i;
        break
    else
        y=10;
    end
end
%Selama antrian virtual terawal tidak penuh maka
paket data yang datang akan memasuki antrian
virtual terawal. Serta, apabila seluruh antrian
virtual penuh maka paket data yang datang akan di-
drop.
```

B.2 Tampilan Simulasi Algoritma Drop-Tail di SimuLink







LAMPIRAN C

C.1 Hasil Simulasi Algoritma AQM

Untuk K = 60

Generated	Simulatio	Volume	Dropped	Packet	Received	Packet-	Total	Jitter
Packets	n Time	Requests	Packets	Time-Out	Packets	Loss	Delay	Juler
418	2372.5187	100	0	17	272	3.2818	331.78	65.059
410	7	100	U	1 /	212	53282	74913	45636
419	2378.9969	100	0	17	273	3.2755	332.49	64.917
419	26	100	U	17	213	29865	16692	7664
516	3087.1310	100	0	26	361	4.2207	421.08	65.465
310	35	100	U	20	301	79221	81988	49267
517	3093.2907	100	0	26	362	4.2139	422.93	65.658
317	8	100	U	20	302	38412	90528	12322
883	5125.6546	200	0	53	601	4.8938	612.71	66.597
003	21	200	U	33	001	13481	75192	44504
884	5132.0641	200	0	54	601	4.9815	613.82	66.755
004	67	200	U	34	001	49815	61704	0272
1374	8303.3557	300	0	102	943	6.0931	1002.0	69.049
1374	04	300	U	102	943	89964	54504	33659
1432	8686.2668	300	0	104	999	6.0046	1055.7	68.897
1432	64	300	0	104	999	18938	92771	81344
1.422	8690.6153	200	0	104	1000	6.0011	1056.6	68.844
1433	31	300	U	104	1000	54068	25864	9988

Generated Packets	Simulation Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
1437	8733.8290	300	0	104	1004	5.9873	1060.2	68.669
1437	22	300	U	104	1004	34485	07754	0917
1438	8738.9231	300	0	104	1005	5.9838	1060.9	68.617
1436	71	300	U	104	1003	89528	96916	78583
2237	13916.011	400	0	154	1649	5.8399	1556.1	67.793
2231	25	400	U	154	1049	69662	66803	40723
2238	13922.430	400	0	154	1650	5.8377	1556.7	67.838
2236	97	400	U	134	1030	55876	36778	04015
2239	13929.434	400	0	154	1651	5.8355	1557.5	67.820
2239	33	400	U	134	1031	43767	28696	78878
2245	13977.904	400	0	155	1656	5.8601	1561.5	67.962
2243	27	400	U	133	1030	13422	45474	38253
2739	16996.151	500	0	199	2016	6.1438	1784.9	68.062
2139	51	300	U	199	2010	71565	79521	88647
2745	17043.709	500	0	200	2021	6.1633	1788.9	68.090
2743	37	300	U	200	2021	28197	25366	57577
3463	21648.117	500	24	329	2566	8.9073	2174.5	66.312
3403	09	300	24	329	2300	93389	20424	72975
4436	27807.940	500	61	560	3256	12.581	2486.8	64.640
4430	83	300	01	300	3230	03728	26485	96242
4771	29906.501	500	61	674	2401	13.944	2566.1	63.971
4//1	51	500	01	674	3491	22311	507	90592

 $Untuk\ K=108$

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
510	3046.669	100	0	26	355	4.2622	418.63	65.949
310	227	100	0	20	333	95082	4605	92467
511	3051.402	100	0	26	356	4.2553	418.52	65.877
311	15	100	0	20	330	19149	36903	70378
516	3087.131	100	0	26	361	4.2207	421.08	65.465
310	035	100	U	20	301	79221	81988	49267
517	3093.290	100	0	26	362	4.2139	422.93	65.658
317	78	100	U	20	302	38412	90528	12322
883	5125.654	200	0	58	601	5.3554	612.71	66.597
003	621	200	Ü	55	001	93998	75192	44504
884	5132.064	200	0	59	601	5.4428	613.82	66.755
004	167	200				04428	61704	0272
1384	8374.865	300	0	103	952	6.1163	1010.6	69.257
1304	071	300	U	103	732	89549	52702	87114
1387	8410.123	300	0	103	955	6.1055	1013.5	69.292
1307	918	300	0	103	755	12745	30376	67298
1433	8690.615	300	0	104	1000	6.0011	1056.6	68.844
1433	331	300	0	104	1000	54068	25864	9988
1437	8733.829	300	0	104	1004	5.9873	1060.2	68.669
1437	022	300	U	104	1004	34485	07754	0917

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
1438	8738.923	300	0	104	1005	5.9838	1060.9	68.617
1436	171	300	U	104	1005	89528	96916	78583
2237	13916.01	400	0	154	1649	5.8399	1556.1	67.793
2231	125	400	U	154	1049	69662	66803	40723
2238	13922.43	400	0	154	1650	5.8377	1556.7	67.838
2238	097	400	0	134	1030	55876	36778	04015
2239	13929.43	400	0	154	1651	5.8355	1557.5	67.820
2239	433				1031	43767	28696	78878
2240	13942.89	400	0 0	154	1652	5.8333	1558.1	67.803
2240	837					33333	24252	87519
2243	13963.84	400	0	154	1655	5.8267	1560.0	67.850
2243	314	400	U	134	1033	1207	41141	88822
2739	16996.15	500	0	199	2016	6.1438	1784.9	68.062
2139	151	300	U	199	2016	71565	79521	88647
2740	16999.69	500	0	199	2017	6.1419	1785.5	68.045
2/40	732	300	U	199	2017	75309	63549	56297
27.45	17043.70	500	0	200	2021	6.1633	1788.9	68.090
2745	937	500	0			28197	25366	57577

200

2025

6.1557

40228

1791.3

82232

68.138

89665

2749

17061.71

95

500

0

C.2 Hasil Simulasi Algoritma Drop-Tail

Untuk K = 60

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
124	787.5936 569	100	0	0	22	0	318.41 87056	58.332 53593
258	1504.858 307	200	0	0	54	0	398.41 86404	58.399 29496
260	1518.984 383	200	0	0	56	0	419.16 72337	63.139 38406
263	1534.621 362	200	0	0	59	0	434.44 21121	58.792 44212
264	1538.912 306	200	0	0	60	0	442.52 42409	59.691 4505
423	2399.874 673	300	0	0	117	0	711.85 00519	68.104 89155
424	2405.664	300	0	0	118	0	717.68	69.188
609	3631.602 868	400	0	0	202	0	941.74 46959	86.275 05188
874	5072.760 043	400	0	55	361	12.971 69811	1625.6 93103	66.424 62213
1177	6989.229 992	500	0	120	548	17.725 25849	2025.8 29259	71.546 92898

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
1 1178	6990.124	500	0	120	549	17.699	2028.9	71.520
	789	200		120	0.,	11504	58479	29551
7133	44766.12	500	500 140	1528	4956	25.146	3280.4	69.191
/133	327	300				99231	69098	65545
7594	47795.08	500	140	1669	5276	25.500	3284.7	66.726
1374	566	300	140			42289	76792	53522
7650	48173.97	500	140	1681	5320	25.468	3285.5	67.229
7650	5	5	140			53147	88863	96206
			•					

Untuk K = 108

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet -Loss	Total Delay	Jitter
114	735.2160 024	100	0	0	13	0	315.12 12535	50.609 70389
115	749.1525 802	100	0	0	14	0	343.02 16223	52.855 56691
116	760.9037 232	100	0	0	15	0	363.34 19759	50.183 53805
232	1337.105 841	200	0	0	30	0	447.72 83377	58.586 70385
233	1346.687 325	200	0	0	31	0	455.56 535	56.945 96097

Generated Packets	Simulatio n Time	Volume Requests	Dropped Packets	Packet Time-Out	Received Packets	Packet- Loss	Total Delay	Jitter
363	2032.286 91	300	0	0	60	0	667.70 98291	54.342 65441
510	3046.669 227	400	0	0	106	4.7620 1333	803.48 54971	71.915 21852
511	3058.772 276	400	0	0	107	4.6811 3487	812.17 32953	72.031 44576
512	3065.671 305	400	0	0	108	4.6402 3819	823.10 70251	72.899 15717
515	3086.627 786	400	0	0	111	6.0135 0165	856.13 92487	69.256 4376
698	4099.455 759	500	0	18	175	9.0909 09091	1199.5 29756	73.180 91423
699	4110.243 766	500	0	19	175	9.5477 38693	1202.9 71969	73.086 34033
1475	9005.675 669	500	0	430	539	44.102 5641	2650.2 89103	68.240 24069
2073	12919.73 999	500	0	738	829	46.916 71964	2848.6 49224	65.770 39204
2190	13657.21 987	500	0	812	873	48.047 33728	2882.1 95346	65.538 74345
3659	23011.00	500	0	1580	1574	50.015	2956.3	66.556

BAB 6 PENUTUP

Setelah rangkaian penentuan parameter dan perancangan yang telah dilakukan maka dapat ditarik kesimpulan. Pembahasan dari bab-bab sebelumnya dan kendala-kendala yang terjadi selama pengerjaan tugas akhir ini akan menjadi bahan pertimbangan atau referensi dalam melakukan penelitian pengembangan dari tugas akhir ini.

6.1 Kesimpulan

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

- 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.

6.2 Saran

Adapun yang disarankan dalam penelitian lanjutan dengan bahasan yang sama, yaitu terkait sistem antrian pada permasalahan *multimedia streaming*, adalah sebagai berikut.

1. Tugas akhir ini melakukan penelitian terhadap sistem antrian pada *multimedia streaming* yang terfokus pada file video,

- sehingga belum dapat digeneralisasikan untuk semua jenis *streaming service*. Harapannya adalah dapat dirumuskan permasalahan secara umum agar dapat diimplementasikan oleh seluruh jenis *multimedia streaming*.
- 2. Pada dasarnya, algortima yang digunakan bernama *Random Early Detection* atau RED. Meskipun algoritma ini tergolong pada salah satu algoritma AQM berbasis antrian, namun algoritma ini tidak dapat digeneralisasikan. Harapannya adalah dapat menspesifikasikan algoritma yang digunakan.
- 3. Permodelan tidak dapat dirancang secara ideal karena jumlah antrian virtual dan kapasitas antrian memiliki nilai yang berubah-ubah. Oleh karena itu, diharapkan pada pengambilan data dilakukan dengan seteliti mungkin sampai dengan penggolongan bobot masing-masing paket data.

DAFTAR PUSTAKA

- [1] S. E. Ghoreishi, A. H. Aghvami dan 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] D. Gross, J. F. Shortie, J. M. Thompson dan C. M. Harris, Fundamentals of Queueing Theory 4th Edition, United States of America: John Wiley & Sons, Inc, 2008.
- [4] M. A. Saleh, H. Hashim dan N. M. Tahir, "A Low Computational Method of Secure Video Streaming in Mobile System," *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2014.
- [5] M. Mathew, "Overview of Temporal Scalability With Scalable Video Coding (SVC)," Texas Instruments Inc, Texas, 2010.
- [6] N. Beheshti, Y. Ganjali, A. Goel dan N. McKeown, "Obtaining High Throughput in Networks with Tiny Buffers," 2008.
- [7] H. Luo, "Improve Delay Performance of Wireless Video Streaming with Active Queue Management," *IEEE*, 2012.
- [8] 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.
- [9] M. D. A. Wibowo, "Analisis dan Implementasi Quality of Service (QoS) Menggunakan IPCOP di SMK Muhammadiyah Imogiri," Naskah Publikasi AMIKOM Yogyakarta, 2014.
- [10] T. 1. 3. V2.1.1, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)," dalam *General aspects of Quality of Service (QoS)*, 1999.
- [11] B. Santosa, "Manajemen Bandwidth Internet dan Intranet," 2004.

RIWAYAT HIDUP



Ilham Fahmi Kurniawan lahir di kota Surabaya pada tanggal 27 Agustus 1994. Putra kedua dan anak ketiga dari tiga bersaudara. Pendidikan dasar ditempuh di SDN 44 Ampenan, kemudian dilanjut di SMPN 7 Mataram dan kemudian di SMAN 15 Surabaya. Pada tahun 2011, penulis menempuh pendidikan tinggi diambil di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Dengan buku tugas akhir ini penulis berharap agar

perjalanan akademik penulis dapat bermanfaat untuk orang banyak.

e-mail: ilham.fahmik94@gmail.com