

# PENGENDALIAN KEPADATAN DALAM *MULTIMEDIA STREAMING DENGAN ACTIVE QUEUE MANAGEMENT*

Oleh :

Ilham Fahmi Kurniawan  
( 2211 100 034 )

Dosen Pembimbing :

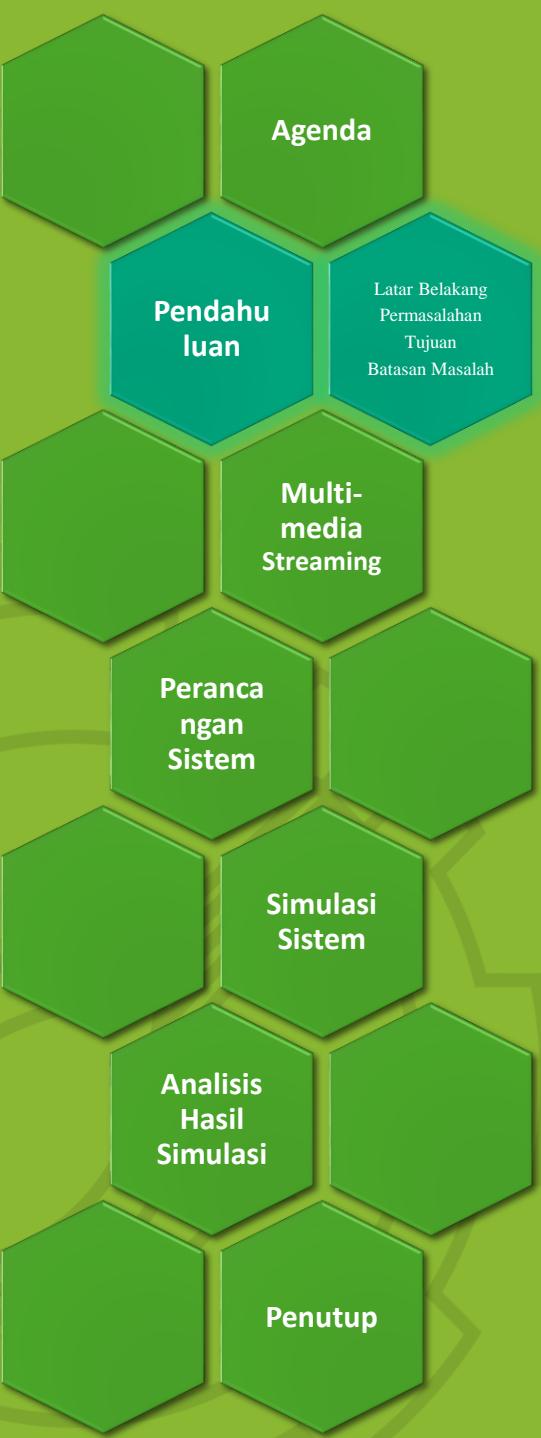
Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.  
Nurlita Gamayanti, S.T., M.T.



# Agenda

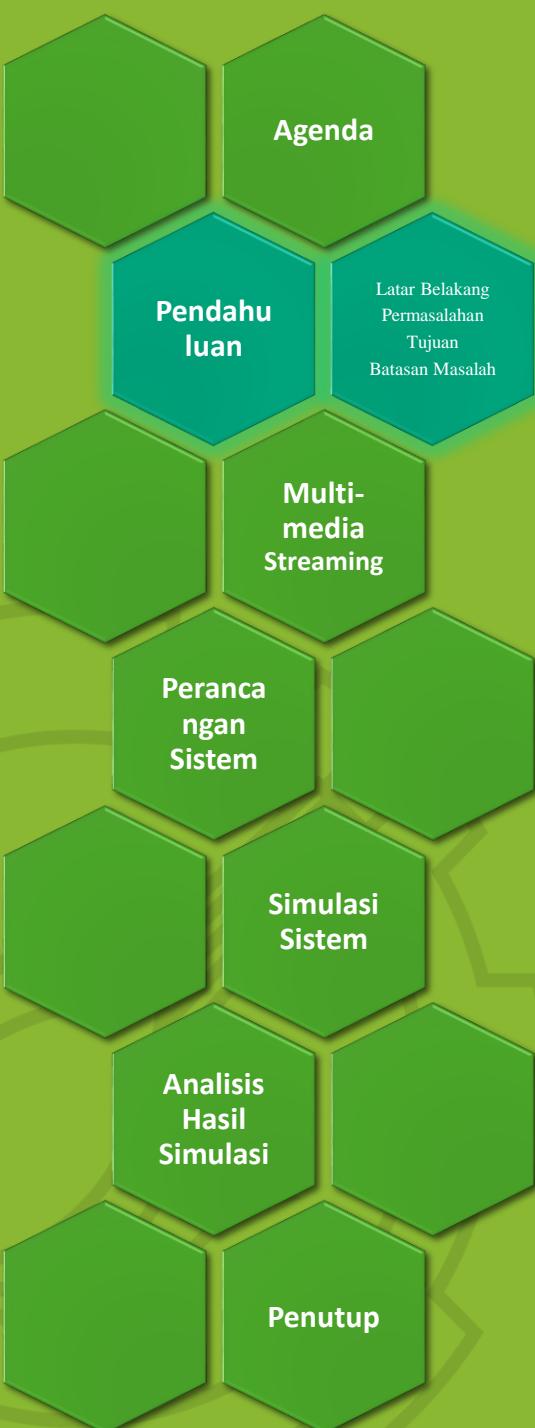
---

- Pendahuluan
- *Multimedia streaming*
- Perancangan sistem
- Simulasi sistem antrian
- Analisis simulasi sistem
- Penutup



# Pendahuluan

- Latar belakang
- Permasalahan
- Tujuan
- Batasan masalah

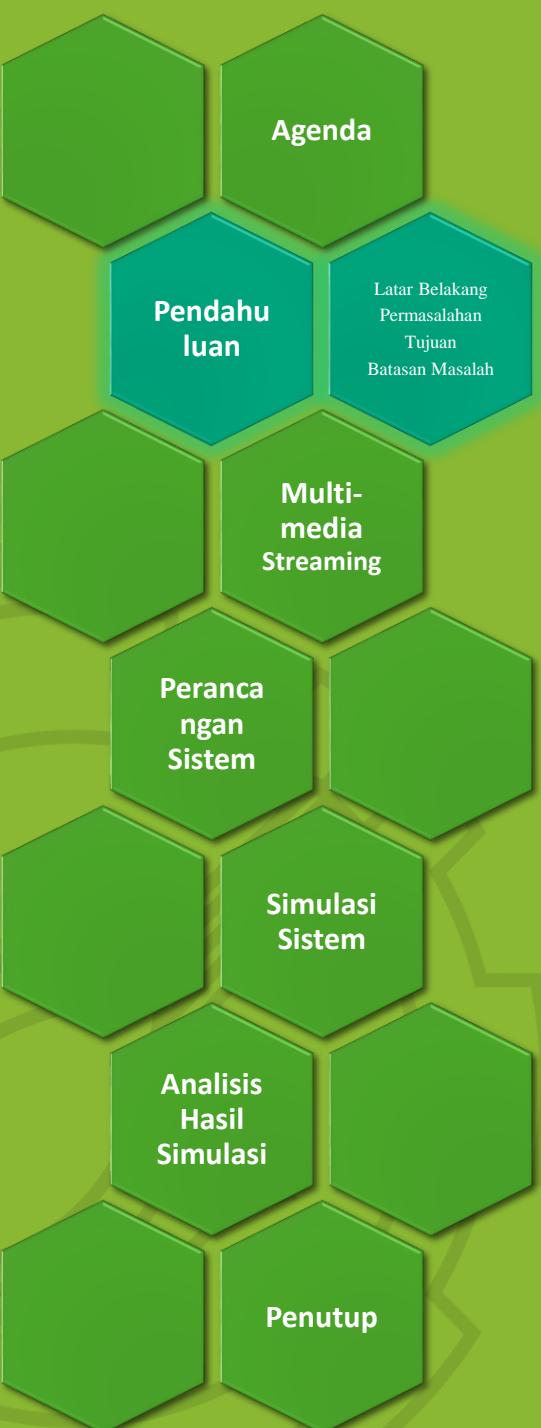


# Pendahuluan

Latar belakang

Peningkatan popularitas ≈ Peningkatan kepadatan



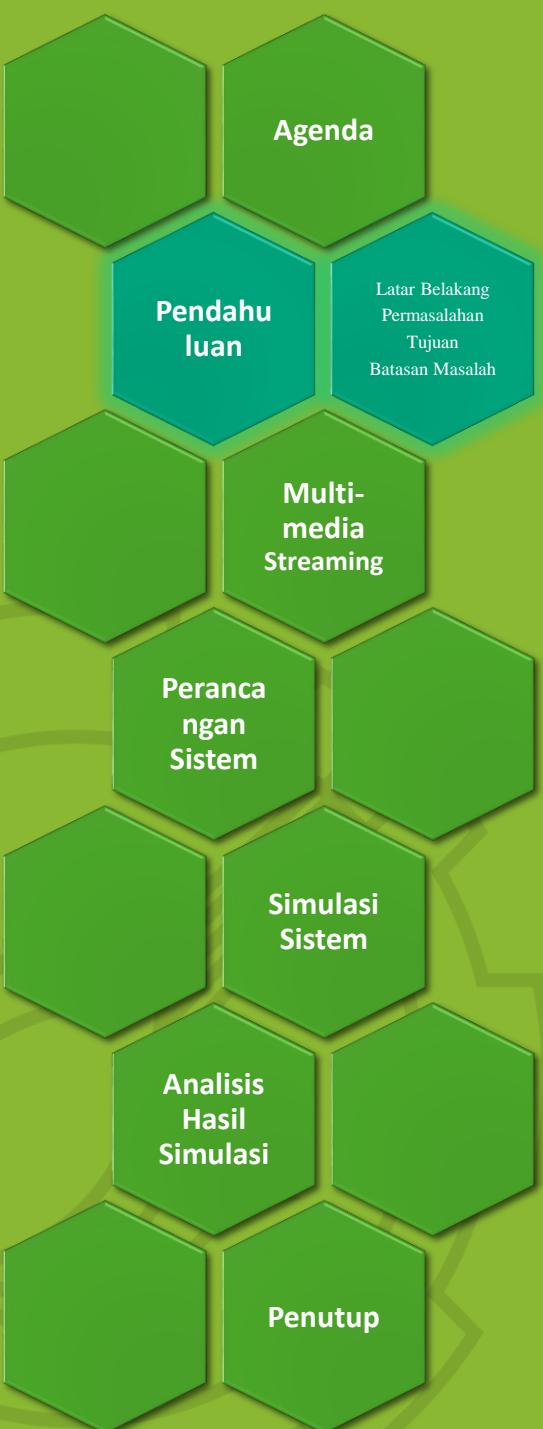


# Pendahuluan

Tujuan

Penelitian ini bertujuan guna menerapkan algoritma AQM berbasis antrian agar mendapatkan QoS yang lebih baik pada *multimedia streaming* pada sistem antrian yang terjadi di *buffer*.

Dimana dalam penelitian ini, QoS yang difokuskan adalah *Packet-loss*, *total delay*, dan *Jitter*.

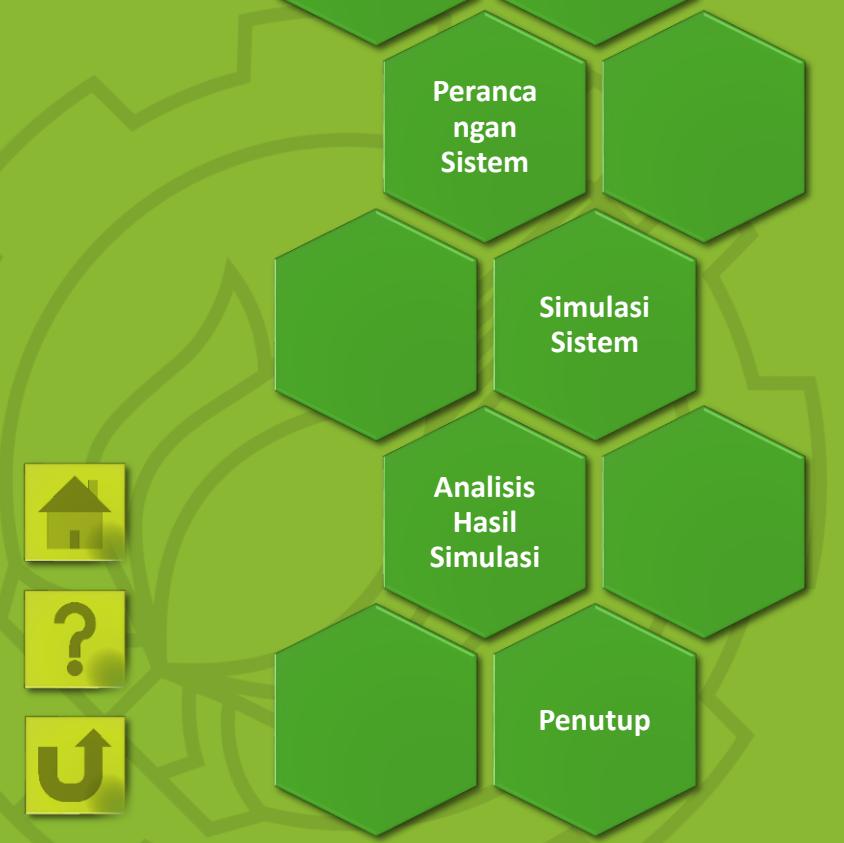


# Pendahuluan

Permasalahan

Permasalahan yang ditemui adalah,

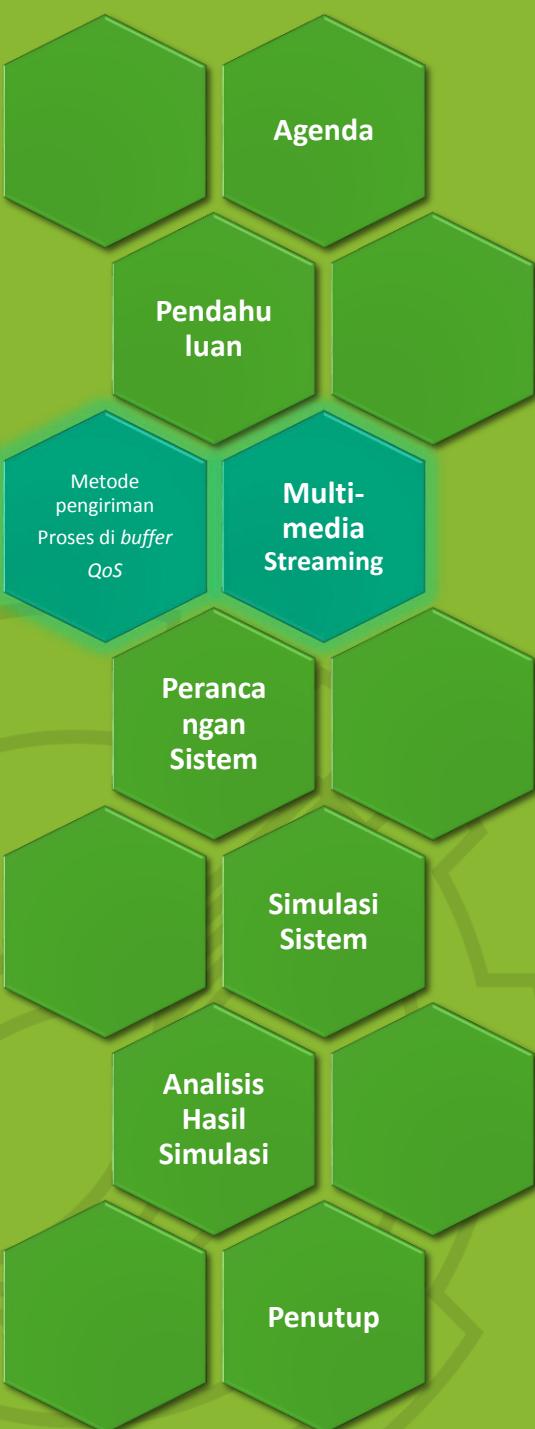
Bagaimana menyelesaikan permasalahan antrian pada *buffer* dengan algoritma *Active Queue Management (AQM)* sehingga QoS yang terukur dapat dibandingkan dengan QoS pada penerapan algoritma drop-tail (metode konvensional).



# Pendahuluan

Batasan masalah

1. Jenis *file* data yang akan digunakan dalam penelitian ini adalah video.
2. Selama proses *multimedia streaming* diasumsikan antrian hanya terjadi pada *buffer*.
3. Ukuran data, kompleksitas data, dan jumlah data pada setiap paket data dalam penelitian ini adalah identik.
4. Kapasitas antrian dan laju pelayanan pada setiap *server* bernilai sama atau identik.
5. Jumlah *server* dalam sistem ditentukan sesuai kebutuhan penelitian.
6. Kualitas dan sinyal jaringan diasumsikan selalu stabil dan dalam kondisi terbaik.



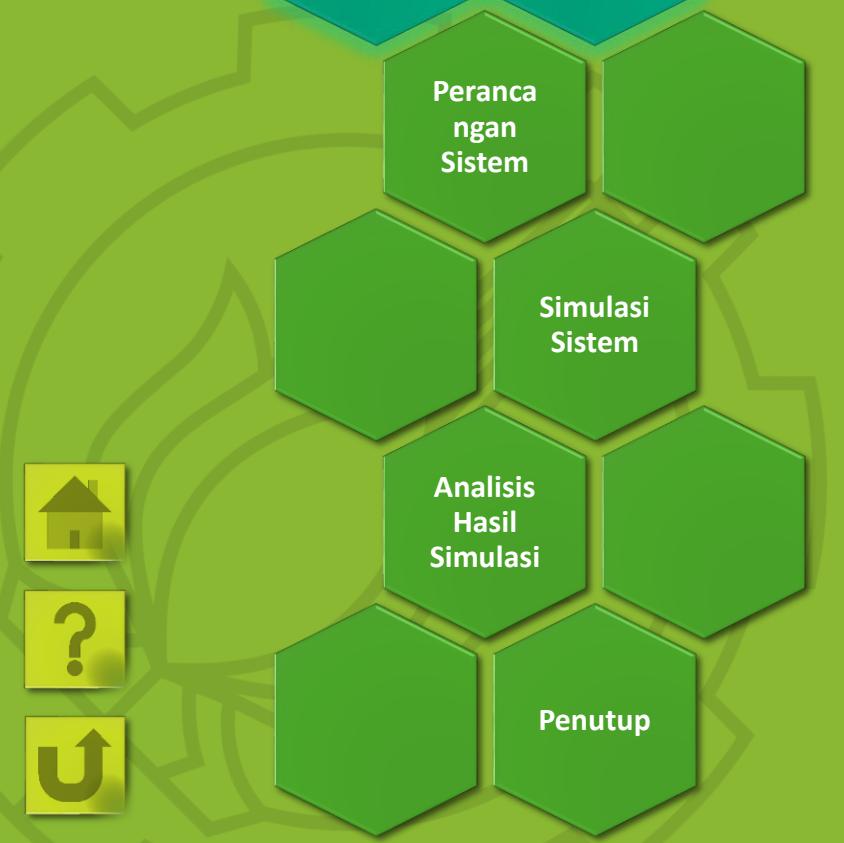
# Multimedia Streaming

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.

*Streaming* adalah teknik mentransfer data agar dapat diproses yang terjadi secara konstan dan berkelanjutan.

Dalam penelitian ini, jenis *file* yang digunakan adalah video.

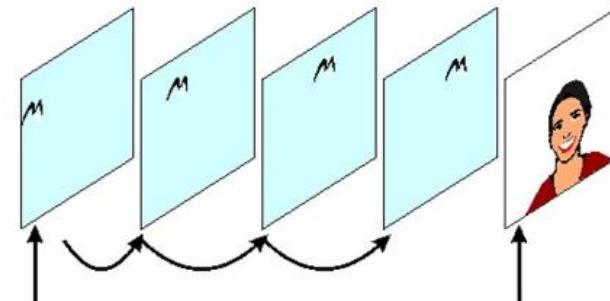
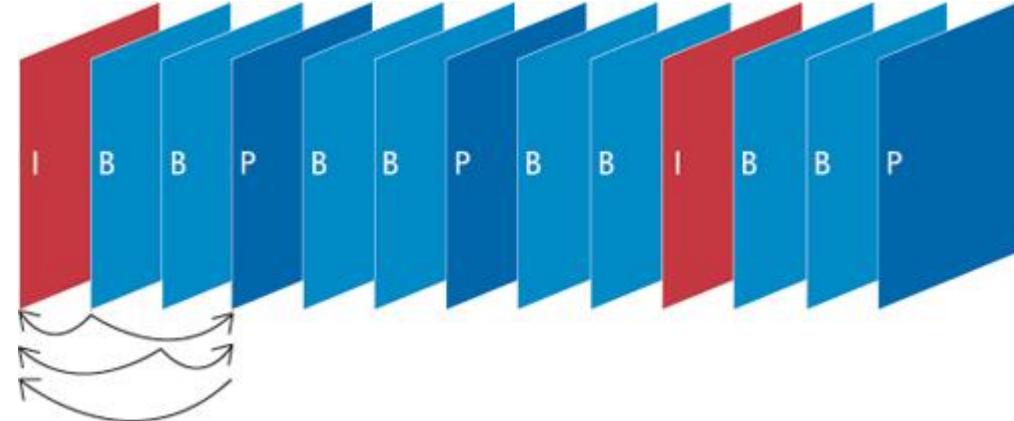


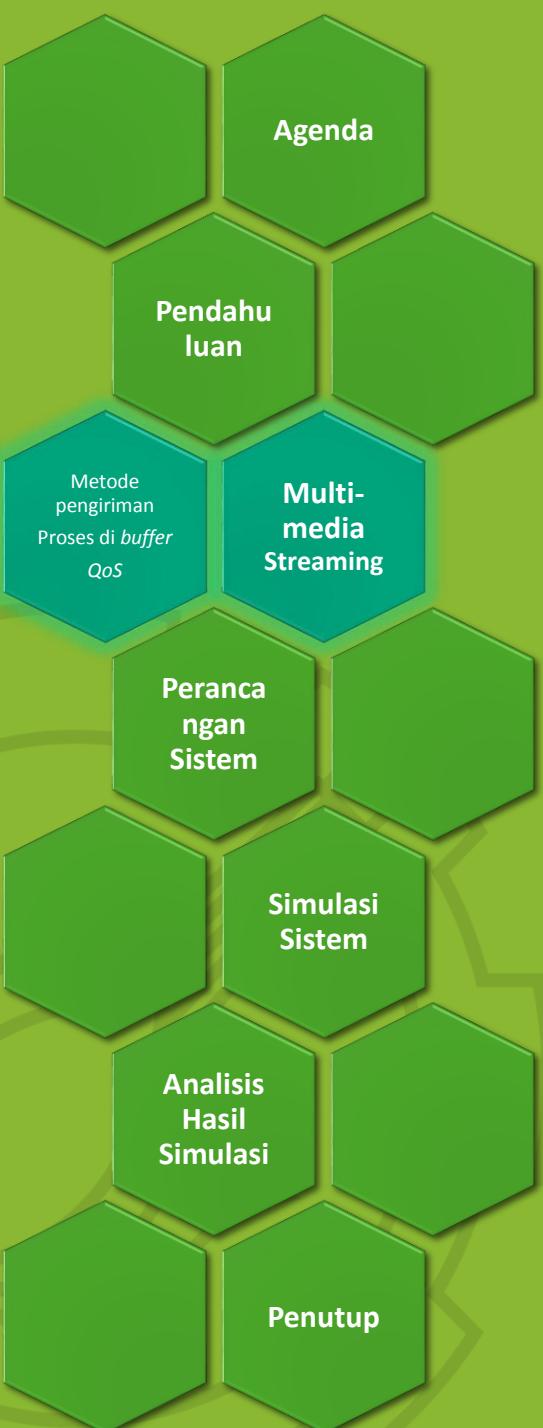


# Multimedia Streaming

Teknik pengiriman data

Secara sederhana adalah dengan membagi video menjadi beberapa bagian paket yang di-*encode* sebelum dikirim dan kemudian paket tersebut di-*decode* agar dapat diputar.





# Multimedia Streaming

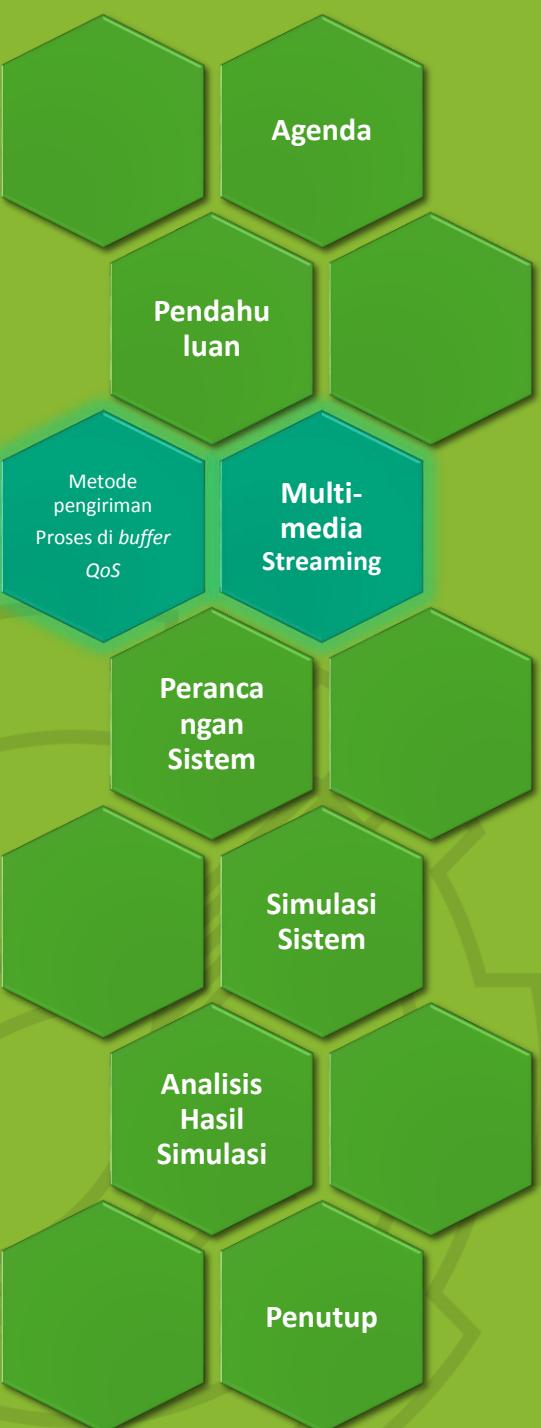
## Proses streaming di buffer

Proses pengiriman paket data *video streaming* yang terjadi pada *buffer*, secara umum dapat digolongkan menjadi 6 tahap.



### Keterangan

1. Proses pengiriman paket diawali oleh pengguna internet yang mengirimkan *request*.
2. *Server* mengakui dan menerima *requests* tersebut.
3. Kemudian jaringan akan terhubung.
4. *File* tersebut akan dibaca sebelumnya oleh *server* sebelum dikirimkan melalui internet.
5. Kemudian *file* tersebut di-write pada *buffer* (dibagi menjadi beberapa bagian dan di-encode)
6. Setelah semua tahap dilakukan, barulah *server* mengirimkan paket data menuju pengguna internet.



# Multimedia Streaming

Proses streaming di buffer

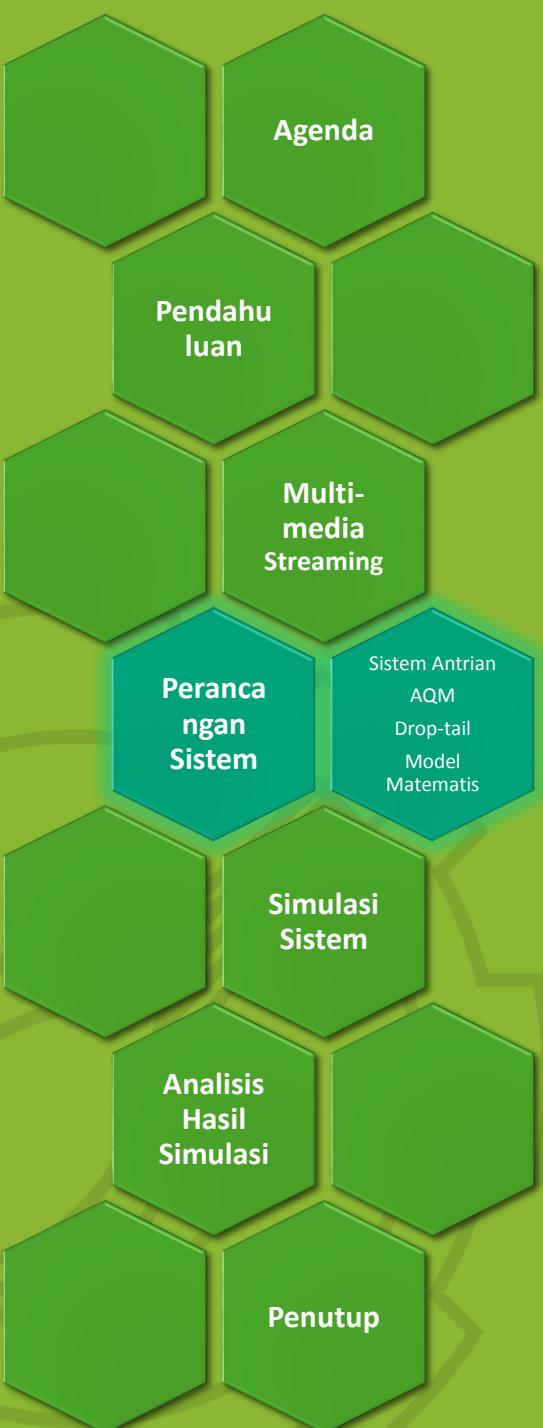
*Quality of Service (QoS)* merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk men-definisikan karakteristik dan sifat dari suatu layanan.

## Komponen-komponen QoS

1. Jitter
2. Bandwidth
3. Latency
4. Packet-loss
5. Troughput

Kategori Jitter	Peak Jitter	Kategori Latency	Besar Delay	Kategori Degradasi	Packet Loss
Sangat Bagus	0 ms	Sangat Bagus	< 150 ms	Sangat bagus	0 %
Bagus	75 ms	Bagus	150 s/d 300 ms	Bagus	3 %
Sedang	125 ms	Sedang	300 s/d 450 ms	Sedang	15 %
Jelek	225 ms	Jelek	> 450 ms	Jelek	25 %

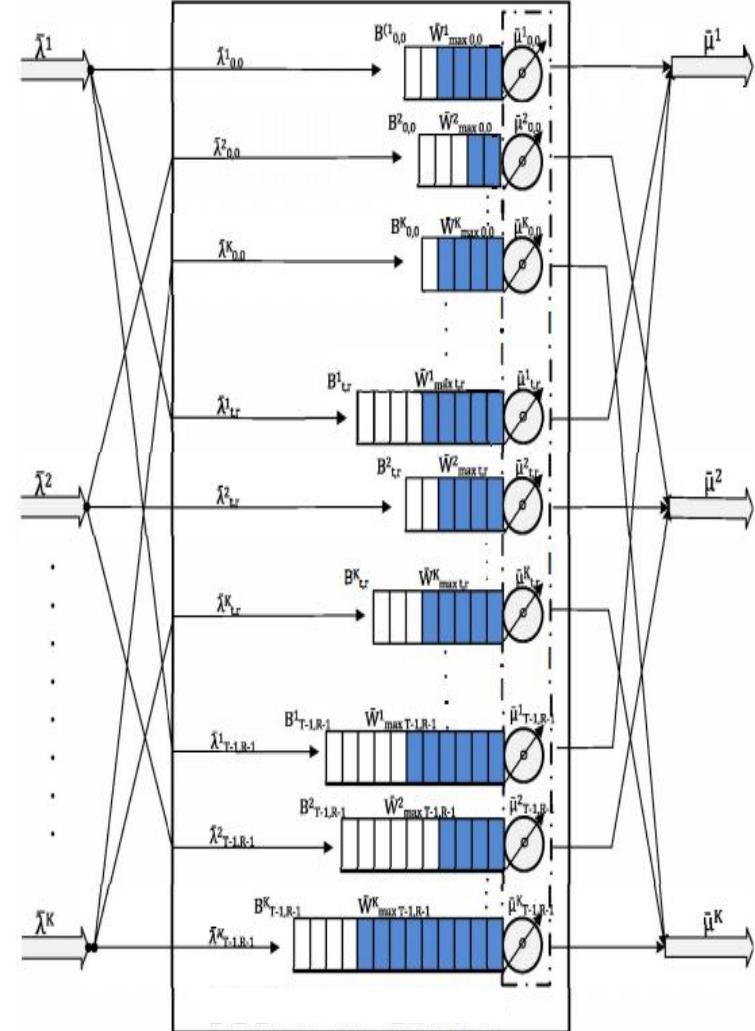
Tabel kategori kualitas layanan



# Perancangan Sistem

Plant utama yang digunakan pada tugas akhir ini merupakan jaringan komputer (meliputi *server* dan *workstation* atau *user*)

Service yang difokuskan pada penelitian ini adalah video streaming

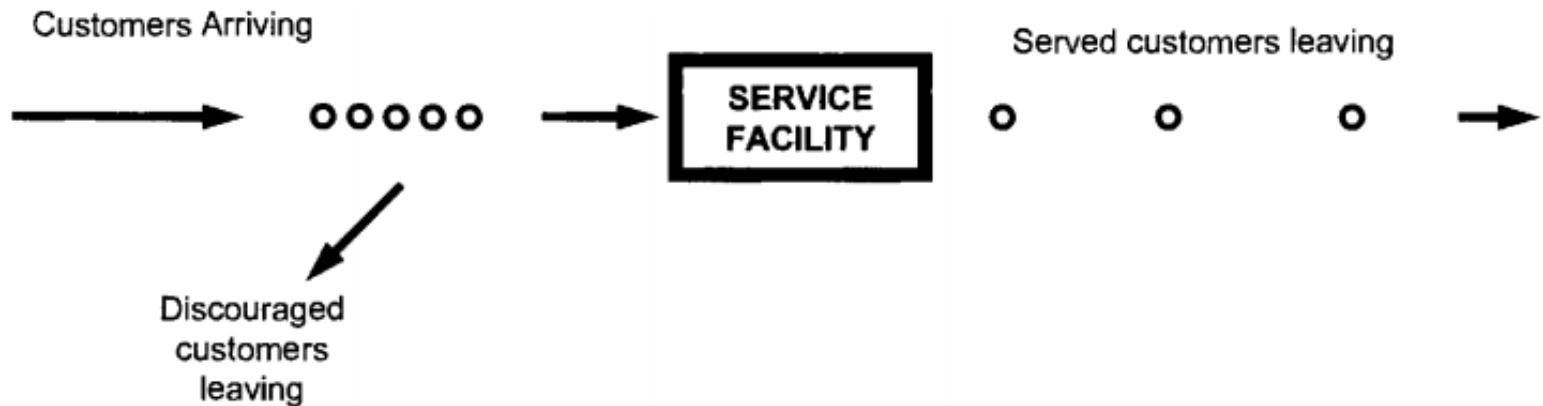


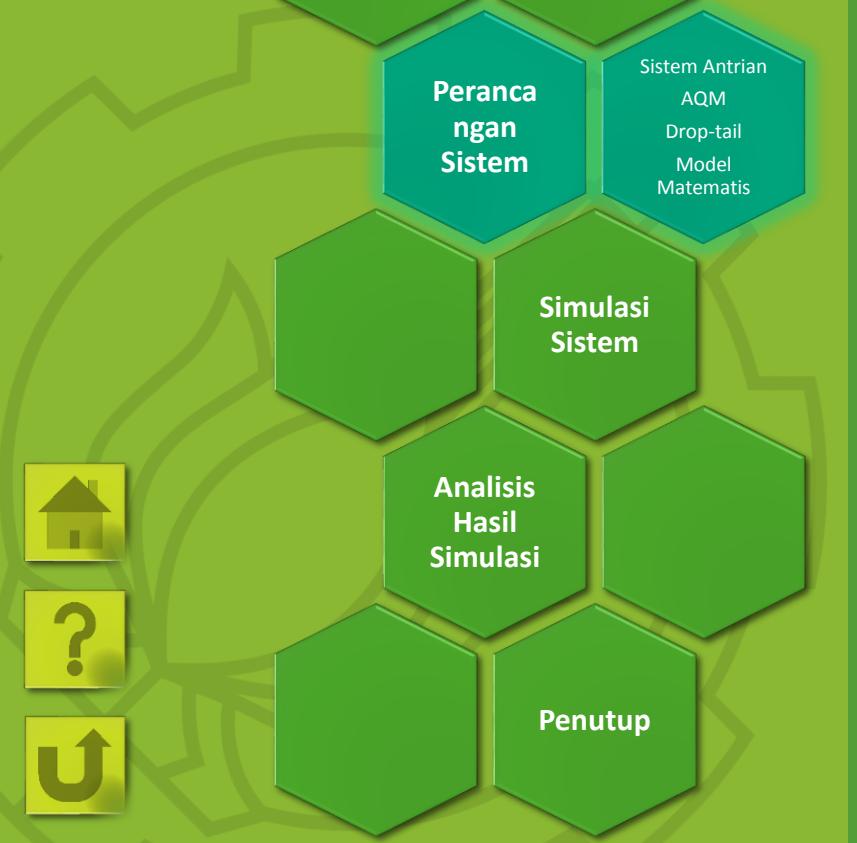


# Perancangan Sistem

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.



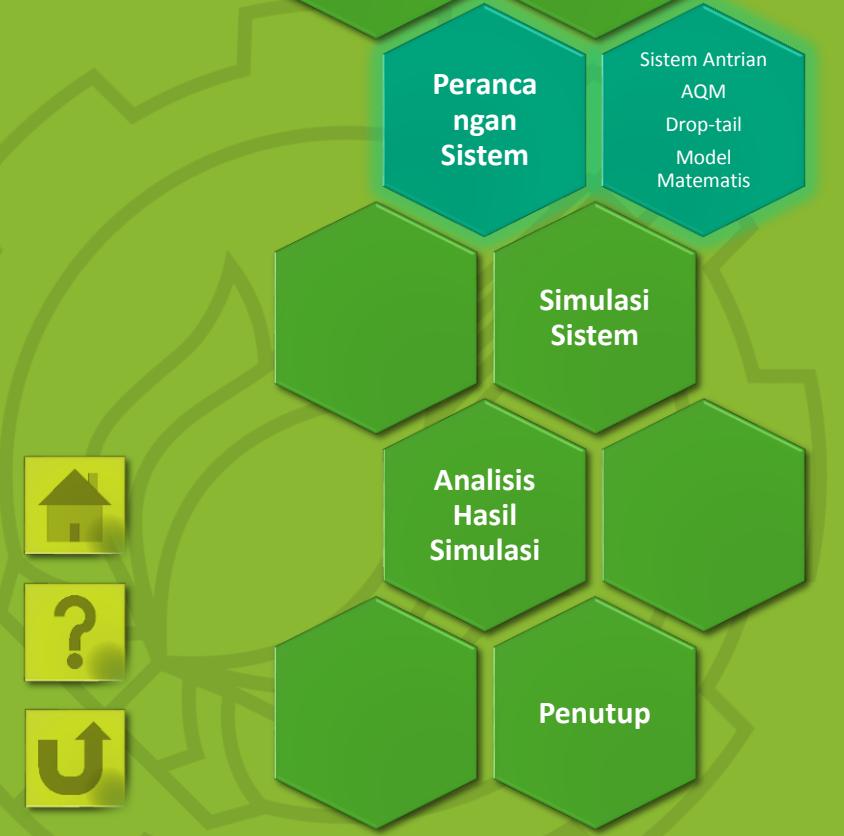


# Perancangan Sistem

Sistem antrian

Performa sistem antrian M/M/c/K adalah sebagai berikut.

Utilisasi server ( $\rho$ )	$\rho = \frac{\lambda}{c\mu}$
Rata-rata jumlah pelanggan dalam antrian ( $L_q$ )	$L_q = \frac{p_0 r^c \rho}{c!(1-\rho)^2} [1 - \rho^{K-c+1} - (1-\rho)(K-c+1)\rho^{K-c}]$
Rata-rata jumlah pelanggan dalam sistem ( $L$ )	$L = L_q + \frac{\lambda_{eff}}{\mu} = L_q + \frac{\lambda(1-P_K)}{\mu}$
Rata-rata lama waktu dalam sistem ( $W$ )	$W = \frac{L}{\lambda_{eff}} = \frac{L}{\lambda(1-P_K)}$
Rata-rata lama waktu dalam antrian ( $W_q$ )	$W_q = W - \frac{1}{\mu} = \frac{L_q}{\lambda_{eff}}$
Proporsi jumlah pelanggan yang ditolak sistem ( $P_K$ )	$P_K = \frac{\lambda^K}{c! c^{(K-c)} \mu^K} P_0$
Menentukan jumlah server ( $c$ )	$c = r + \Delta \quad ; \quad \Delta > 0$

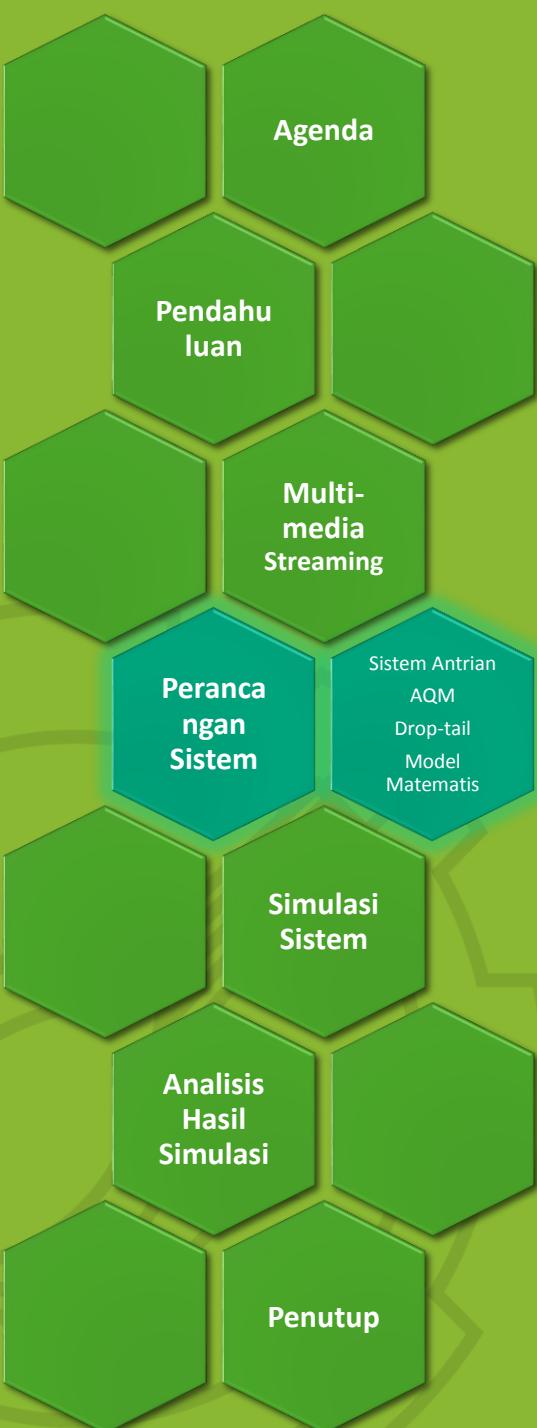


# Perancangan Sistem

Sistem antrian

Keterangan :

- $\rho$  = Utilisasi server
- $W$  = Lama waktu dalam sistem (s)
- $W_q$  = Lama waktu dalam antrian (s)
- $L$  = Jumlah pelanggan dalam sistem
- $L_q$  = Jumlah pelanggan dalam antrian
- $\lambda_{eff}$  = Laju kedatangan efektif (/s)
- $c$  = Jumlah server
- $r$  = Beban kerja antrian
- $P_k$  = Proporsi pelanggan yang ditolak sistem
- $K$  = Kapasitas antrian
- $P_0$  = Probabilitas tidak ada pelanggan
- $\lambda$  = Laju kedatangan (/s)
- $\mu$  = Laju pelayanan (/s)
- $\Delta$  = Jumlah server tambahan pada kondisi beban kerja berlebih (dapat berupa bilangan desimal untuk membulatkan jumlah server)



# Perancangan Sistem

## Algoritma AQM

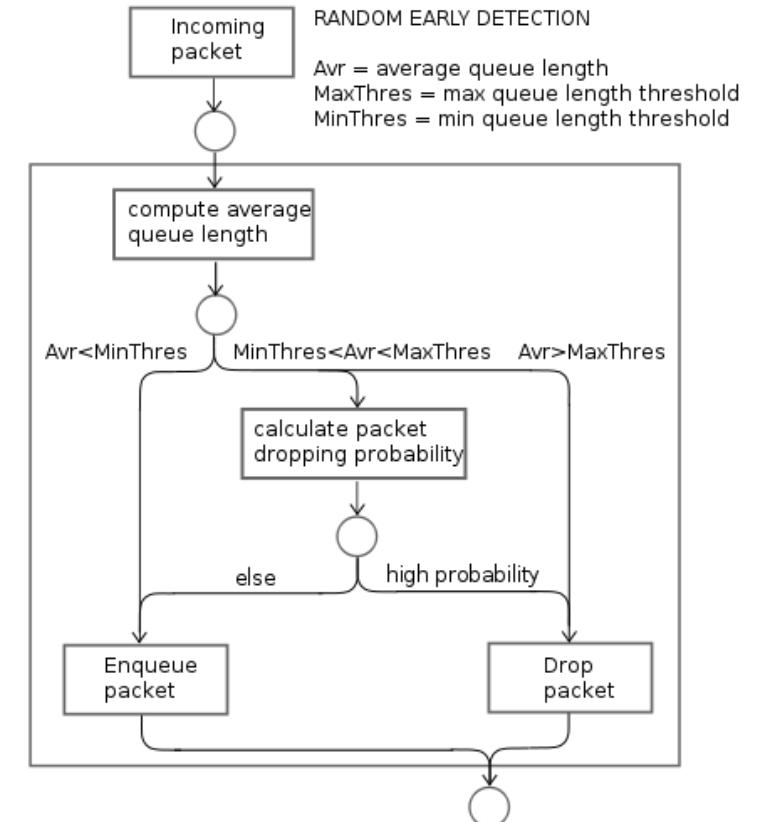
Pendekatan *Active Queue Management* (AQM) secara umum dibagi menjadi 2 kategori, yaitu AQM berbasis antrian dan AQM berbasis beban

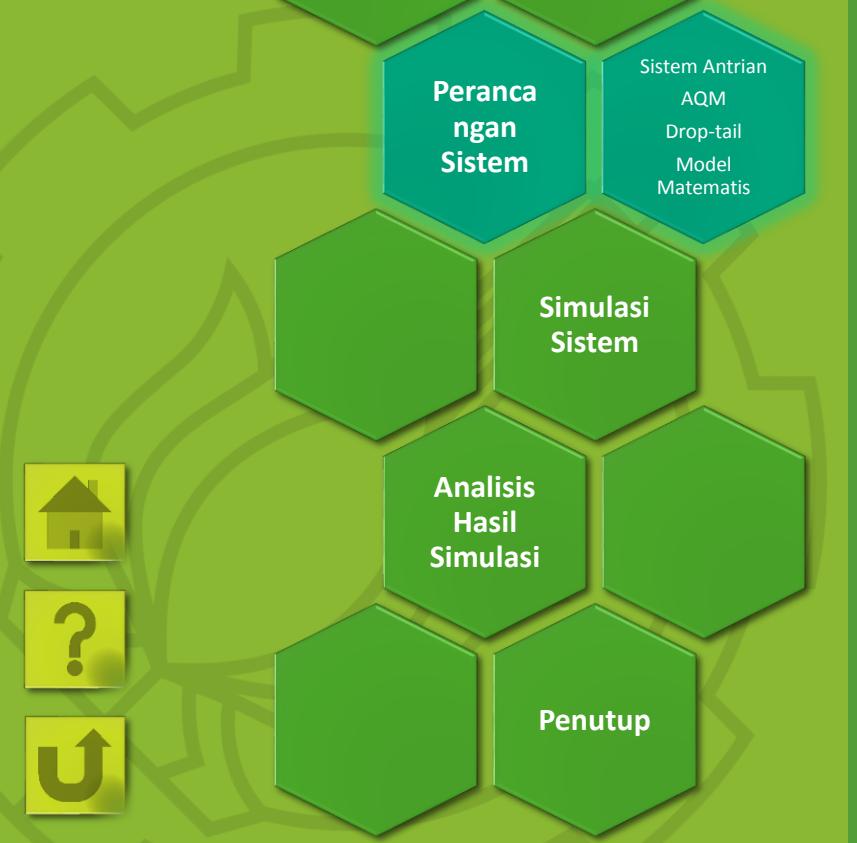
### AQM berbasis antrian

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. Beban dapat didefinisikan menjadi perbedaan antara laju kedatangan dan laju pelayanan.



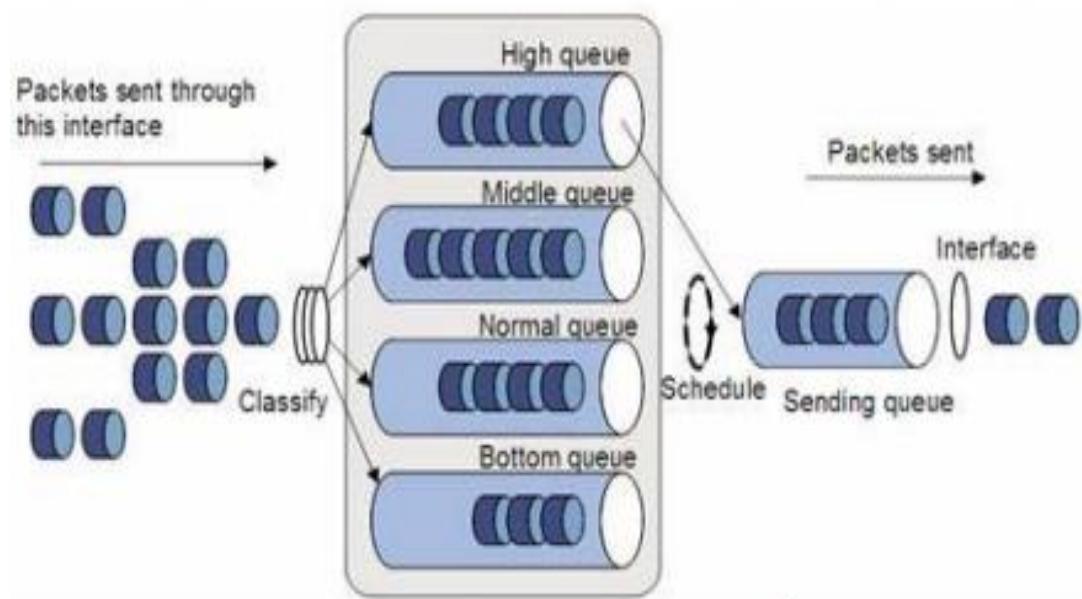


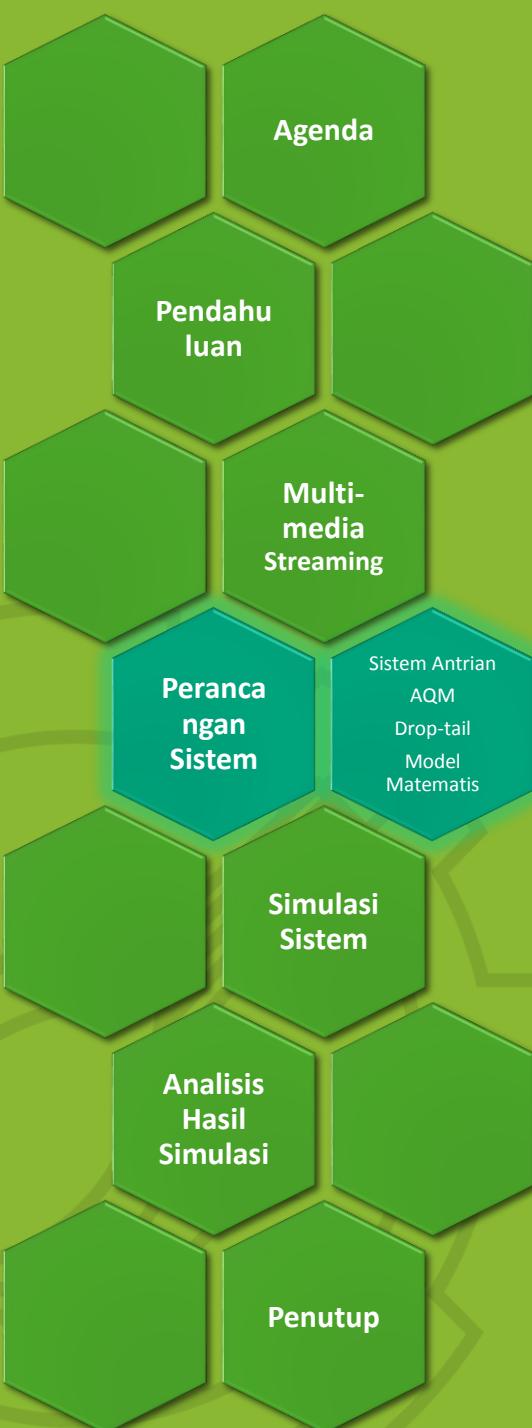
# Perancangan Sistem

## Algoritma drop-tail

*Drop-Tail Queue Management* adalah algoritma manajemen antrian sederhana yang digunakan pada *router*, yang mana akan meniadakan paket berdasarkan pada panjang antrian.

Drop-Tail memiliki dua karakteristik, yaitu lock-out dan full queue

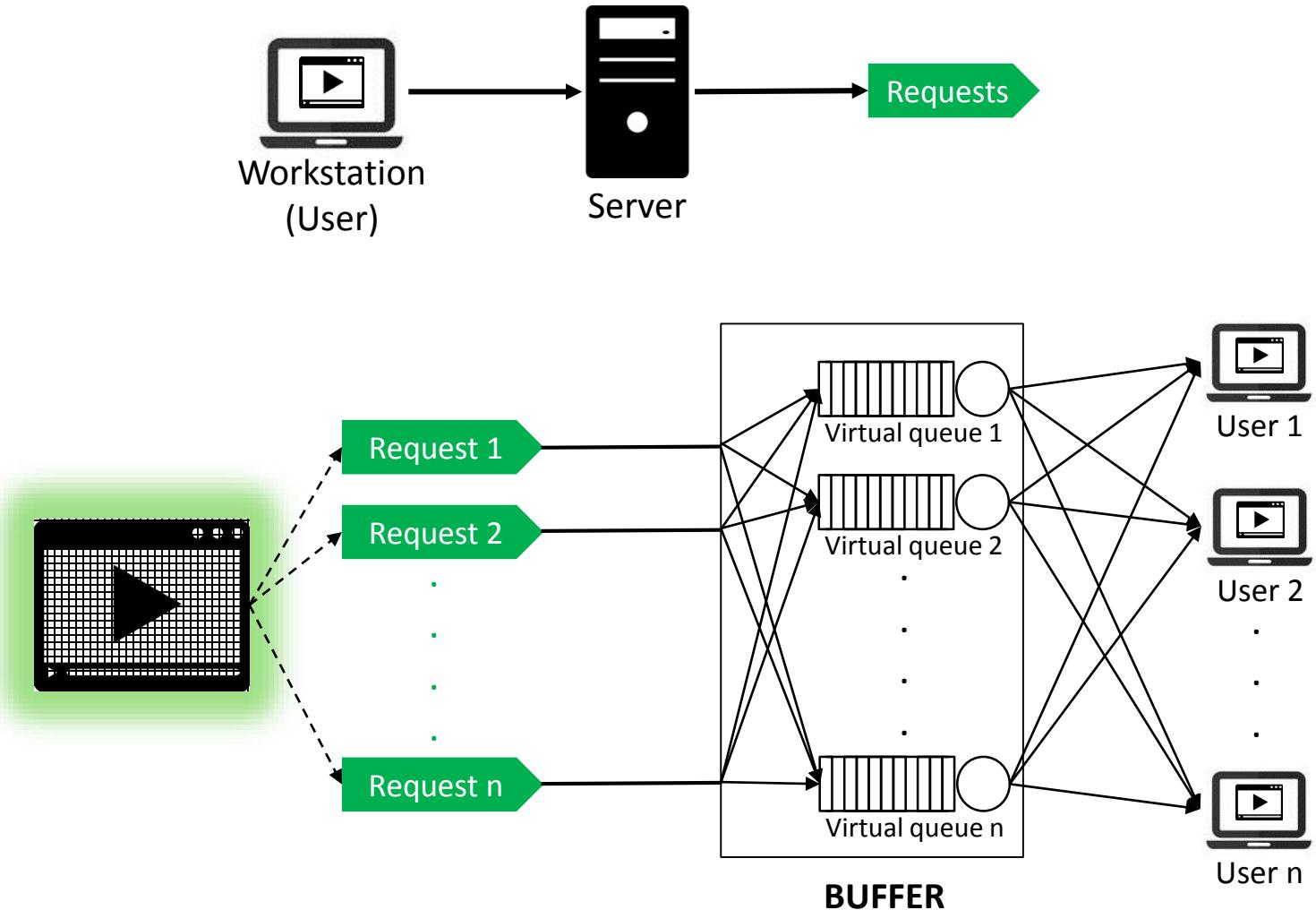


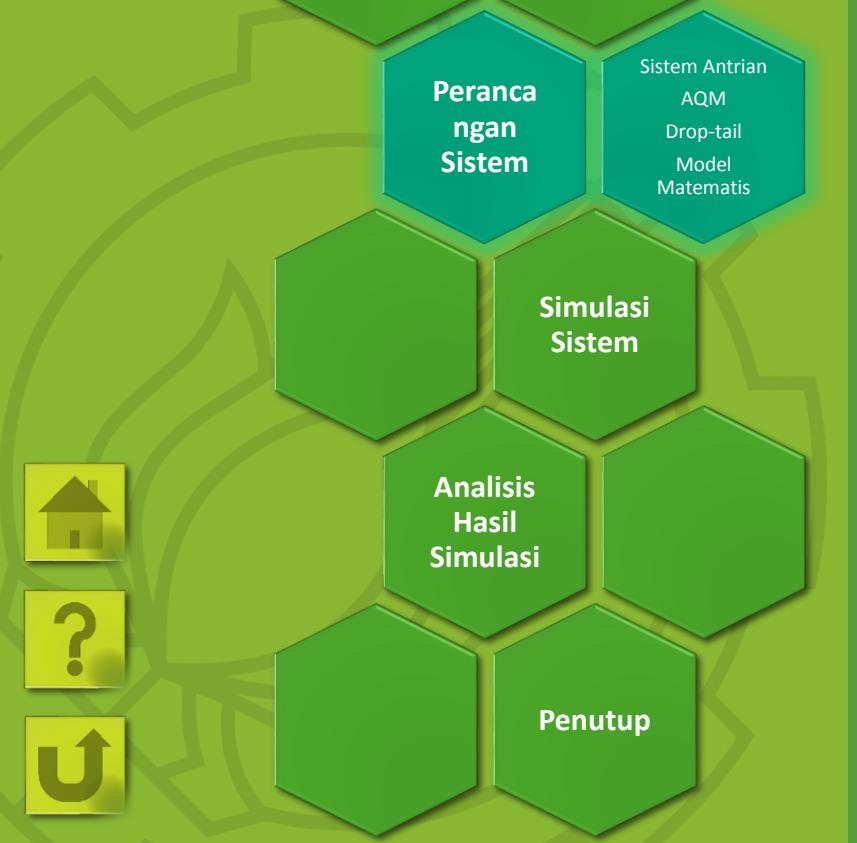


# Perancangan Sistem

Model matematis

Model fisik dan skenario sistem antrian pada *buffer*





# Perancangan Sistem

Model matematis

## Fungsi objektif

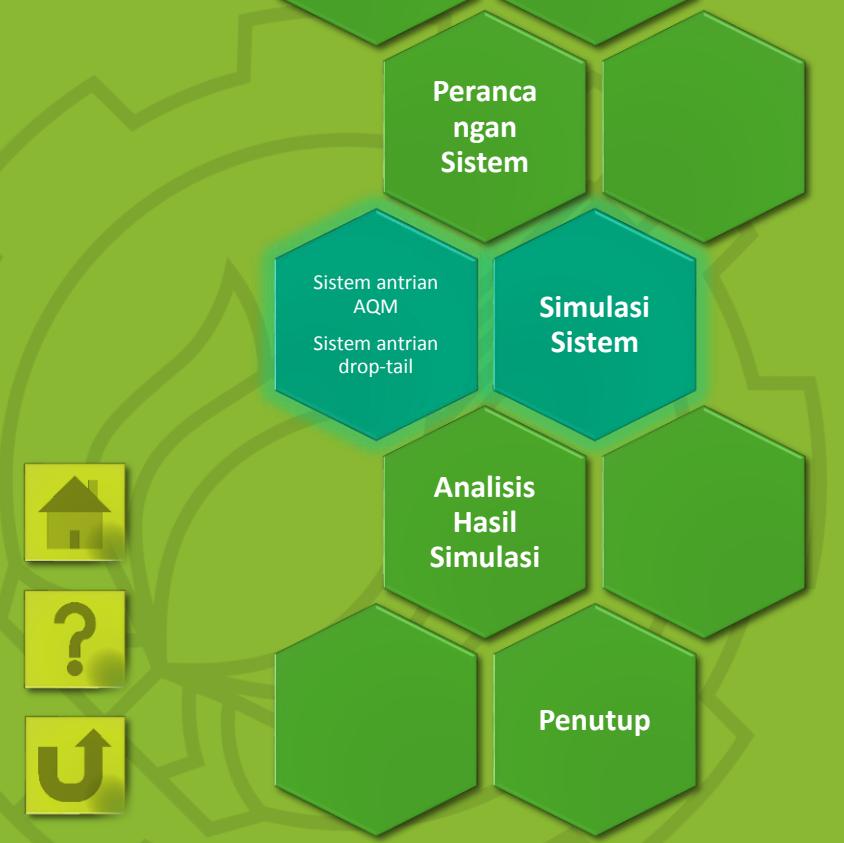
$$\min \sum_{n=1}^c \bar{W}_{q_n} + 1/\bar{\mu}_n$$

## Fungsi kendala

1.  $\bar{W}_n \leq \bar{W}_{max}$
2.  $\frac{1}{\bar{\mu}_n} \leq \frac{1}{\bar{\mu}_{max}}$
3.  $\sum_{n=1}^c K_n \leq K_{max}$

### Keterangan :

$\bar{W}$	= Rata-rata waktu paket data di dalam sistem
$\bar{W}_{max}$	= Rata-rata waktu <i>time-out</i> paket data di dalam sistem
$\bar{\mu}$	= Rata-rata laju pelayanan <i>server</i>
$\bar{\mu}_{max}$	= Rata-rata laju paket yang <i>time-out</i> pada pelayanan
$K$	= Kapasitas antrian pada masing-masing <i>server</i>
$K_{max}$	= Kapasitas fisik <i>buffer</i>
$\bar{W}_q$	= Rata-rata waktu paket data di dalam antrian
$\bar{\mu}$	= Rata-rata laju pelayanan <i>server</i>



# Simulasi Sistem

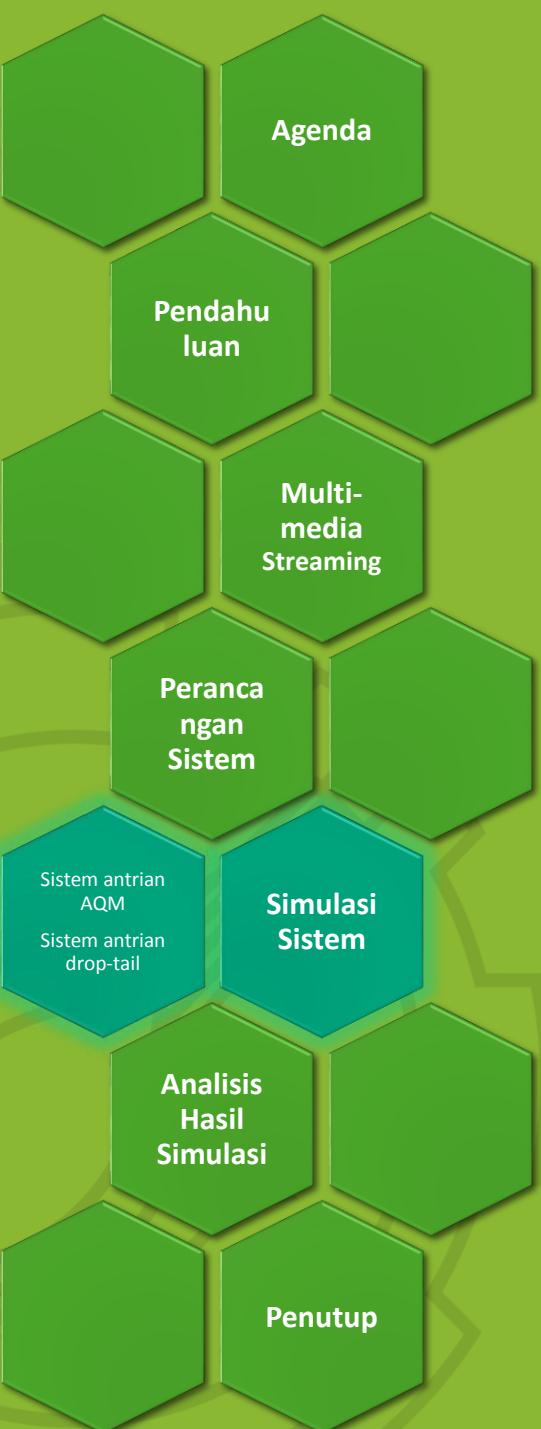
Inisialisasi data diambil dari server localhost, yang dinamai *Kirby*.

(<http://10.122.1.99/>)

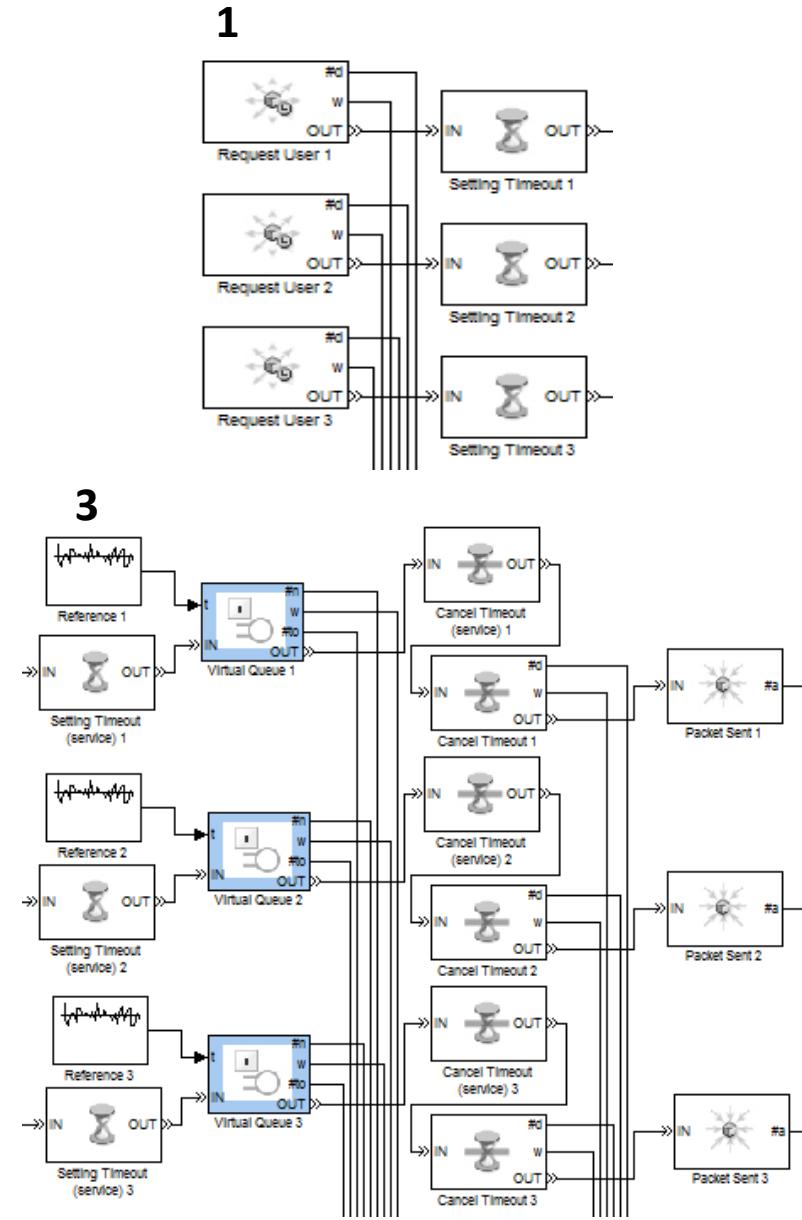
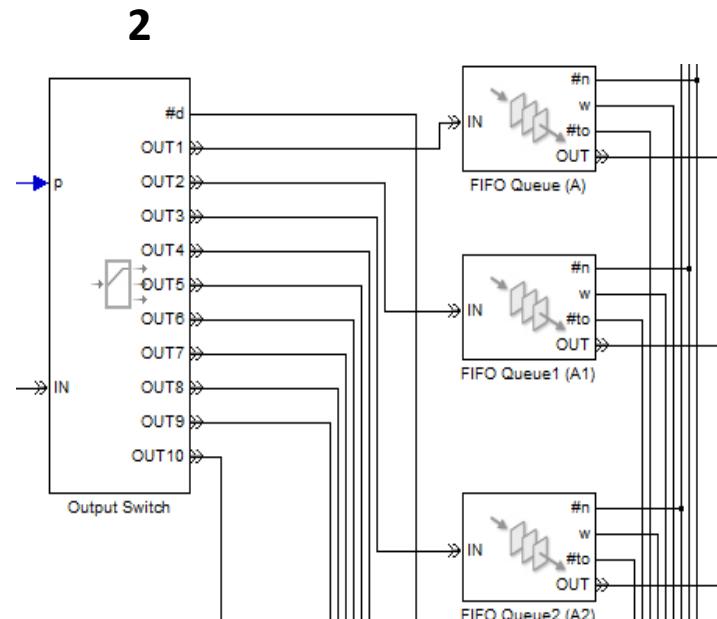
Spesifikasi Server Kirby	
<b>Processor</b>	Intel Dual Core E6800 series
<b>Memory</b>	4 GB
<b>Storage</b>	HDD 150 GB
<b>OS</b>	Ubuntu 14.04.3 LTS
<b>Ethernet</b>	1 Gbps
<b>Webserver</b>	Nginx/1.4.6 Ubuntu

Hasil olahan data adalah sebagai berikut.

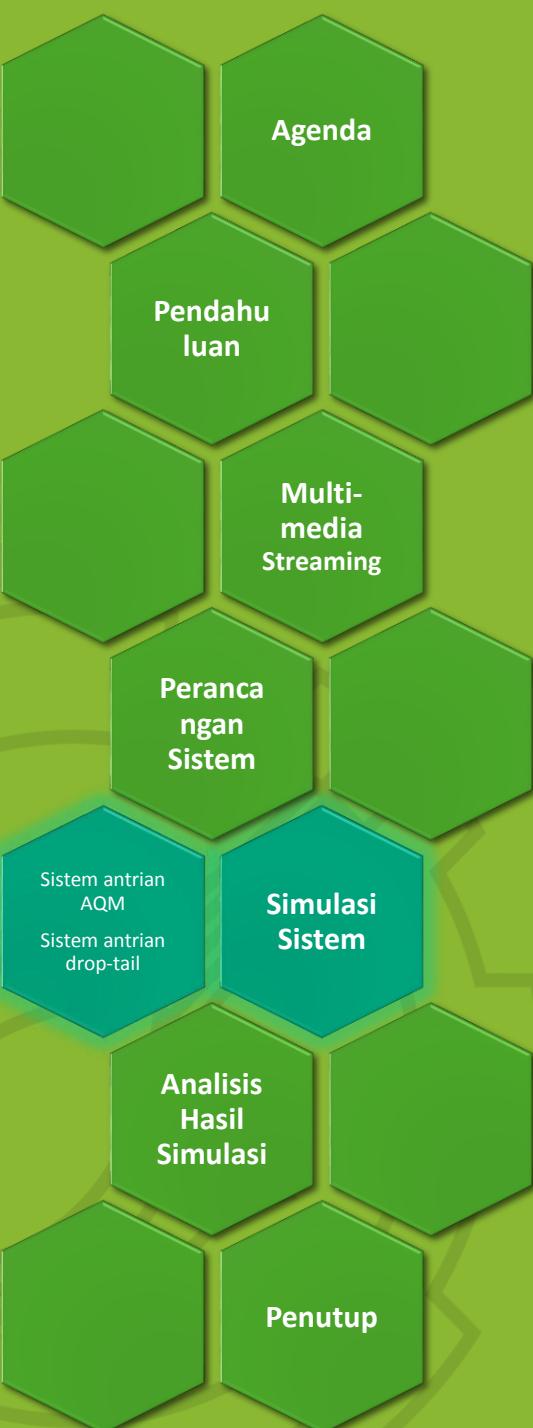
Nilai Parameter Data Inisial		
$1/\lambda_n$	= Waktu antar kedatangan	= 81
$1/\mu_n$	= Waktu pelayanan setiap antrian virtual	= 76
$1/\mu_{max}$	= Waktu time-out paket data	= 3630
$c$	= Jumlah server	= 9
$K_n$	= Kapasitas antrian virtual	$\frac{24000}{c * (\text{paket data})}$



# Simulasi Sistem



Algoritma AQM



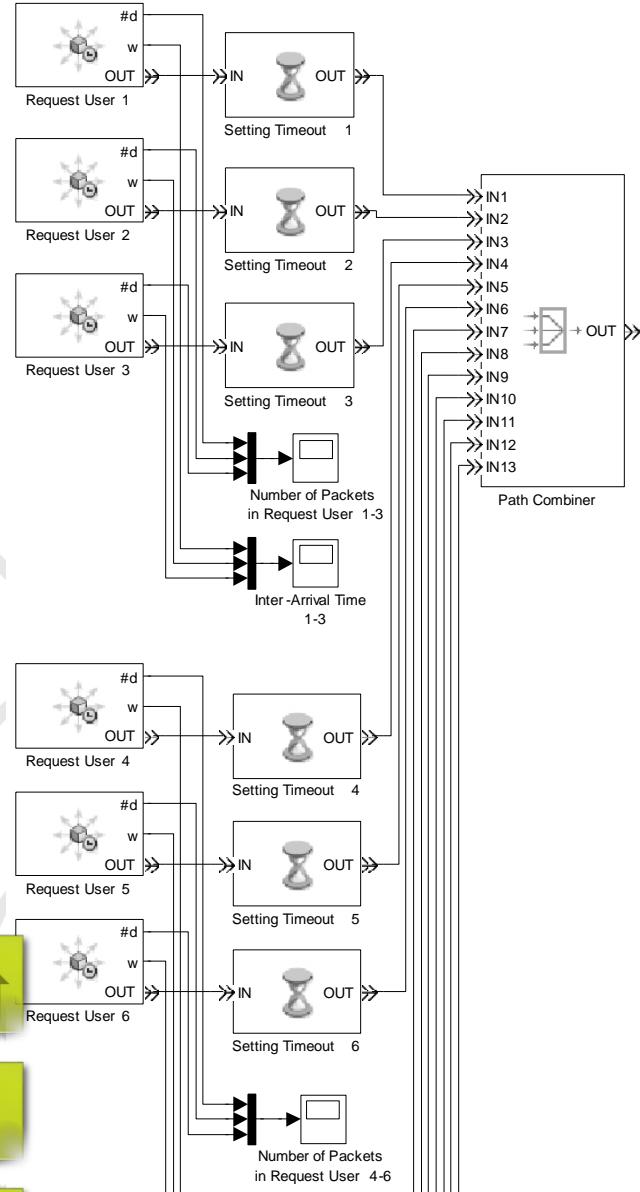
# Simulasi Sistem

Algoritma AQM

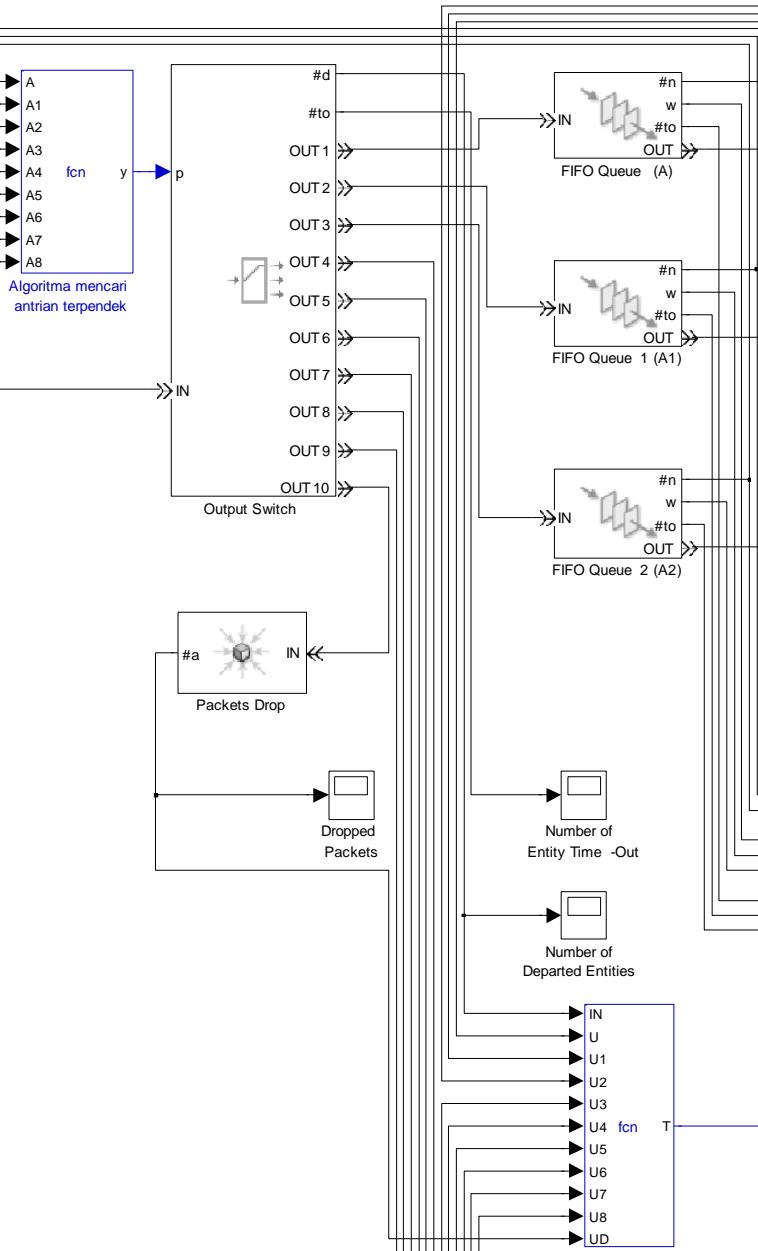
```
1 function y = fcn(A, A1, A2, A3, A4, A5, A6, A7, A8)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5
6 %Algoritma untuk mencari panjang antrian terpendek.
7 - N = [A A1 A2 A3 A4 A5 A6 A7 A8];
8 - T = [50 50 50 50 50 50 50 50 50];
9 - R = T - N;
10 - [B,C] = max(R);
11 - if max(B) <= 0
12 -     y = 10;
13 - else
14 -     y = C;
15 end
```

```
1 function T = fcn(IN, U, U1, U2, U3, U4, U5, U6, U7, U8)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5
6 %Untuk menentukan nilai Time-out berdasarkan Time-out.
7 - K = 100 * (sum([U U1 U2 U3 U4 U5 U6 U7 U8])) / IN;
8 - T = 90 + (10 * K);
9 - if IN == 0
10 -     T = 90;
11 - elseif T >= 225
12 -     T = 224;
13 end
```

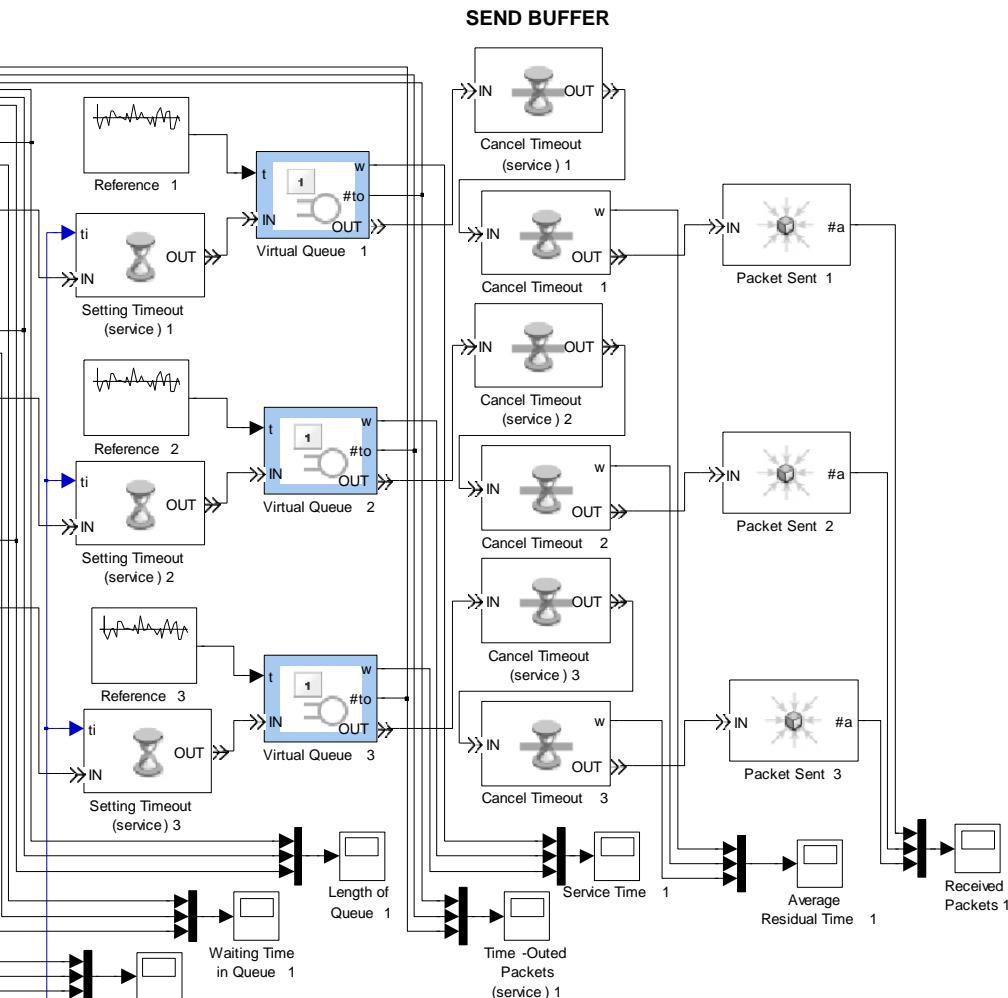
## READ FILE

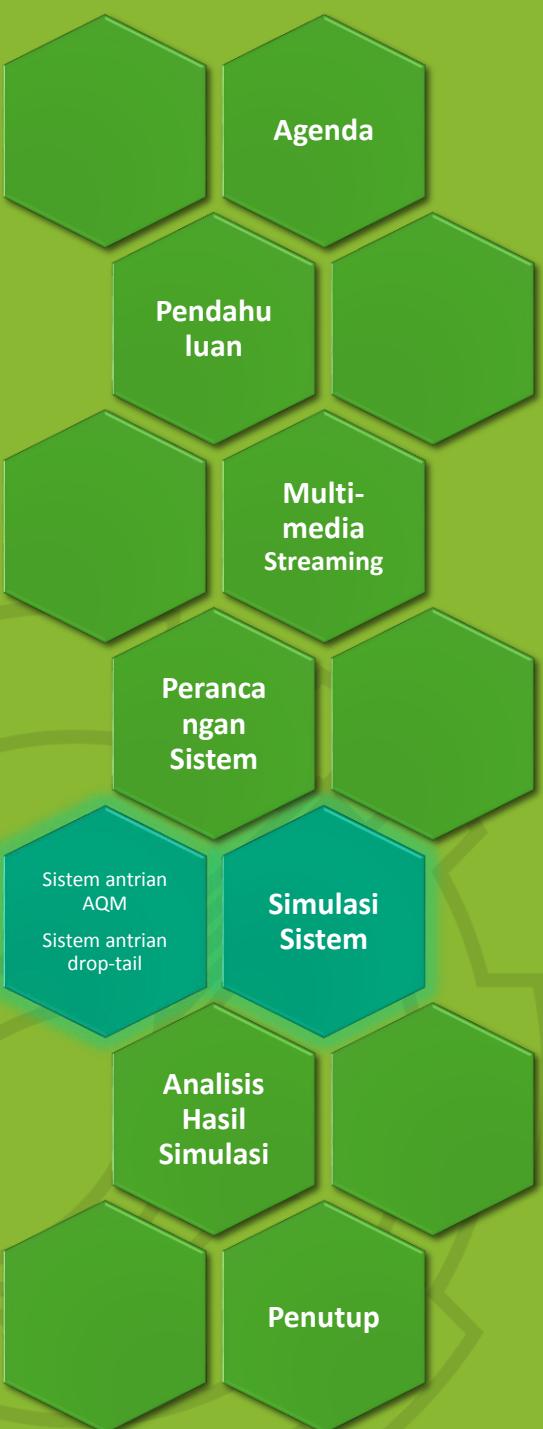


## WRITE BUFFER



## SEND BUFFER

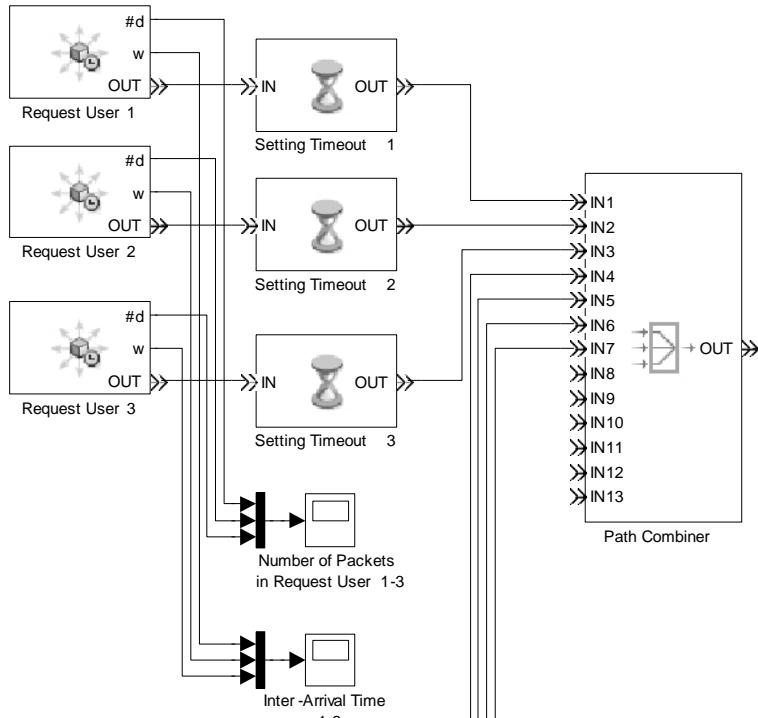
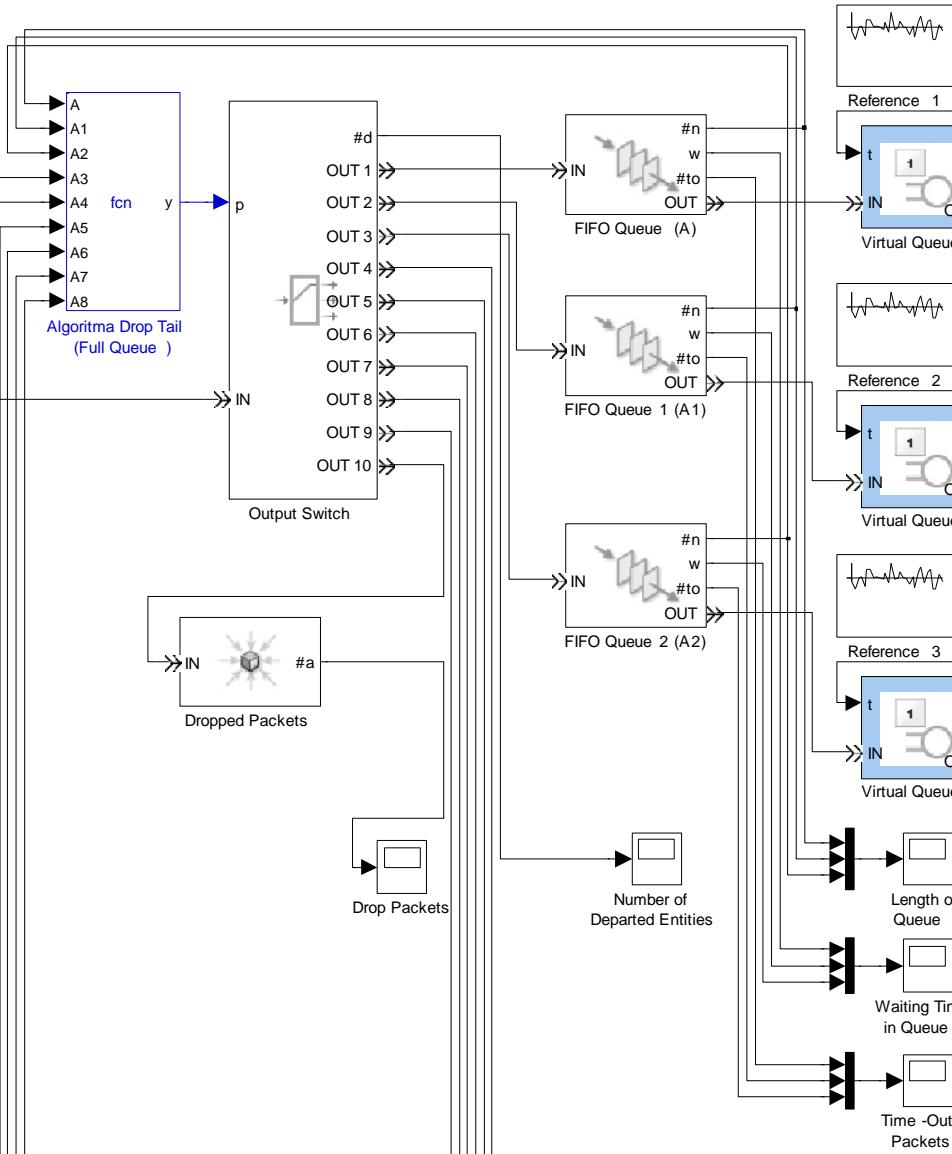
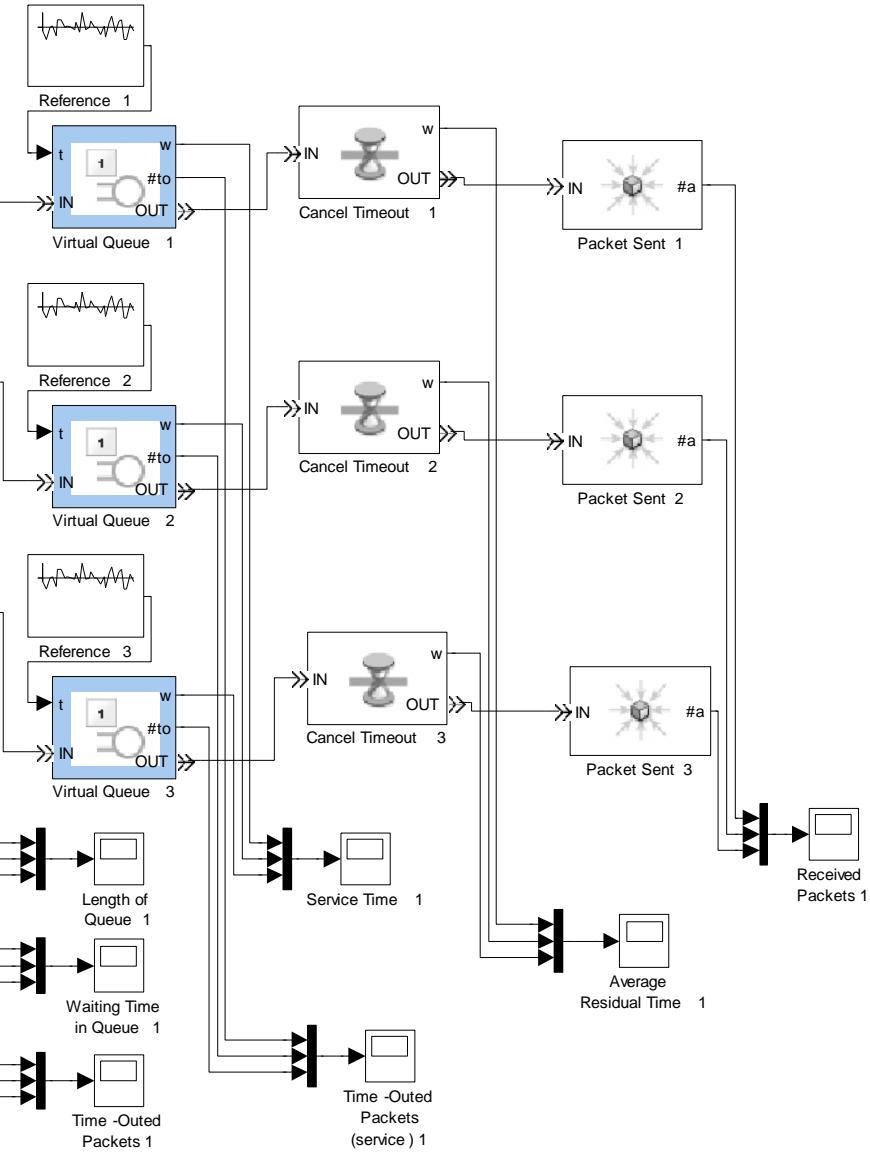


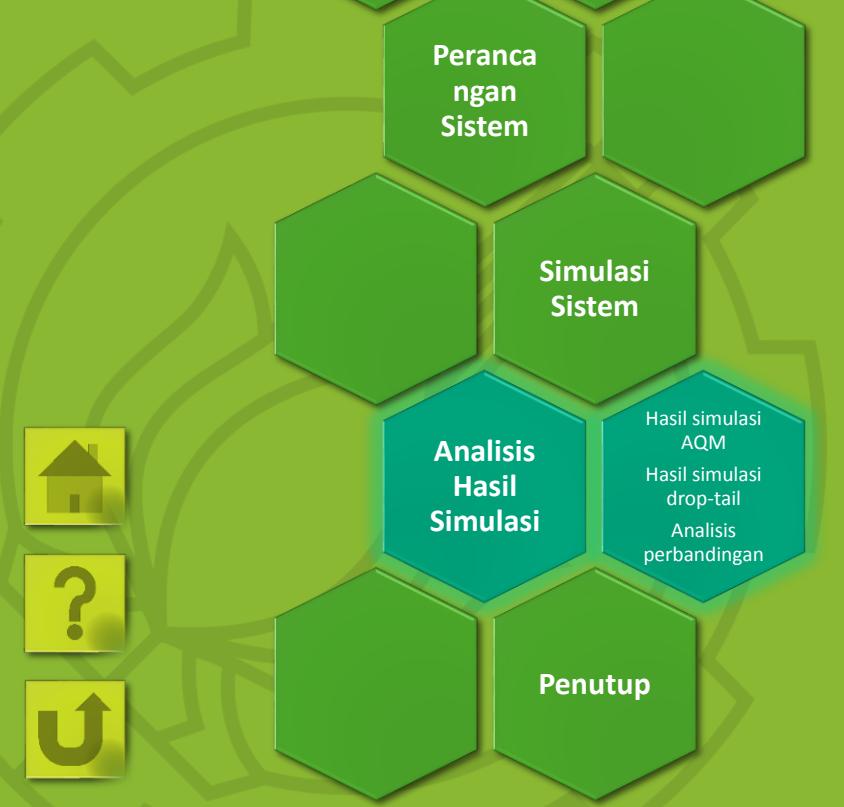


# Simulasi Sistem

Algoritma drop-tail

```
1 function y = fcn(A, A1, A2, A3, A4, A5, A6, A7, A8)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5
6 %Algoritma untuk Lock-out dan Full queue
7 - N = [A A1 A2 A3 A4 A5 A6 A7 A8];
8 - T = [50 50 50 50 50 50 50 50 50];
9 - R = T-N;
10 - y=0;
11 for i=1:9
12 - if R(i)>0
13 - y=i;
14 - break
15 - else
16 - y=10;
17 end
18 end
```

**READ FILE****WRITE BUFFER****SEND BUFFER**



# Analisis Hasil Simulasi

Pengujian dilakukan dengan memberikan asumsi besarnya masing-masing paket data.

1. Asumsi nilai paket data sebesar 25 Kb. Sehingga,

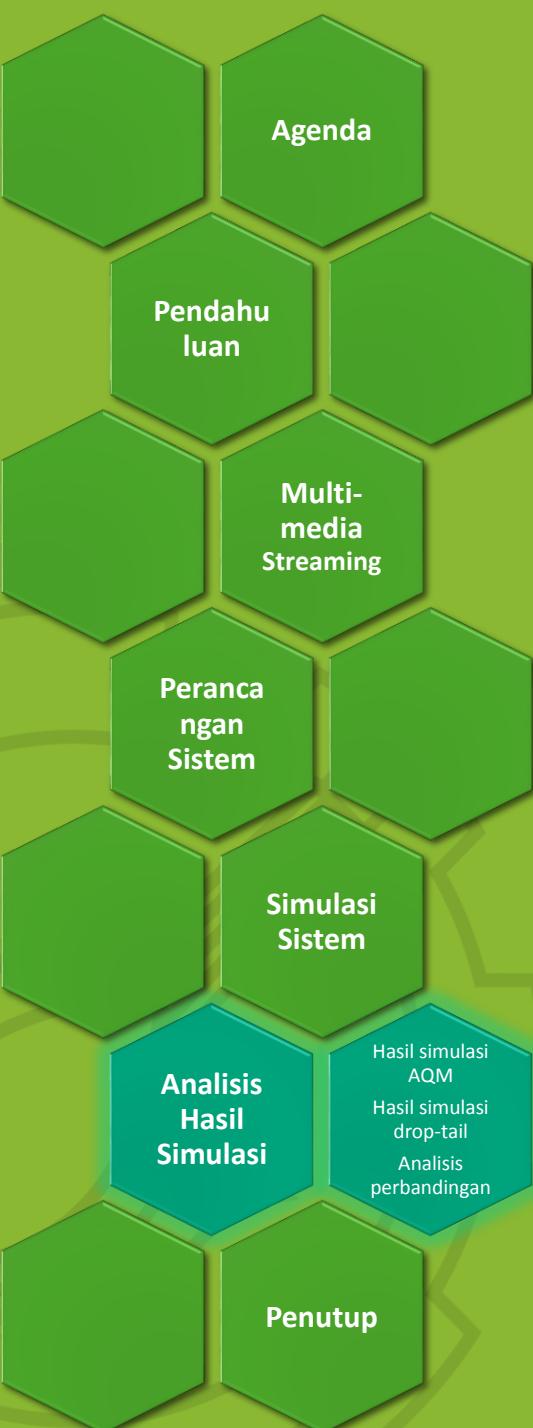
$$K_{\max} = 972$$

$$\underline{K = 108}$$

2. Asumsi nilai paket data sebesar 45 Kb. Sehingga,

$$K_{\max} = 540$$

$$\underline{K = 60}$$



# Analisis Hasil Simulasi

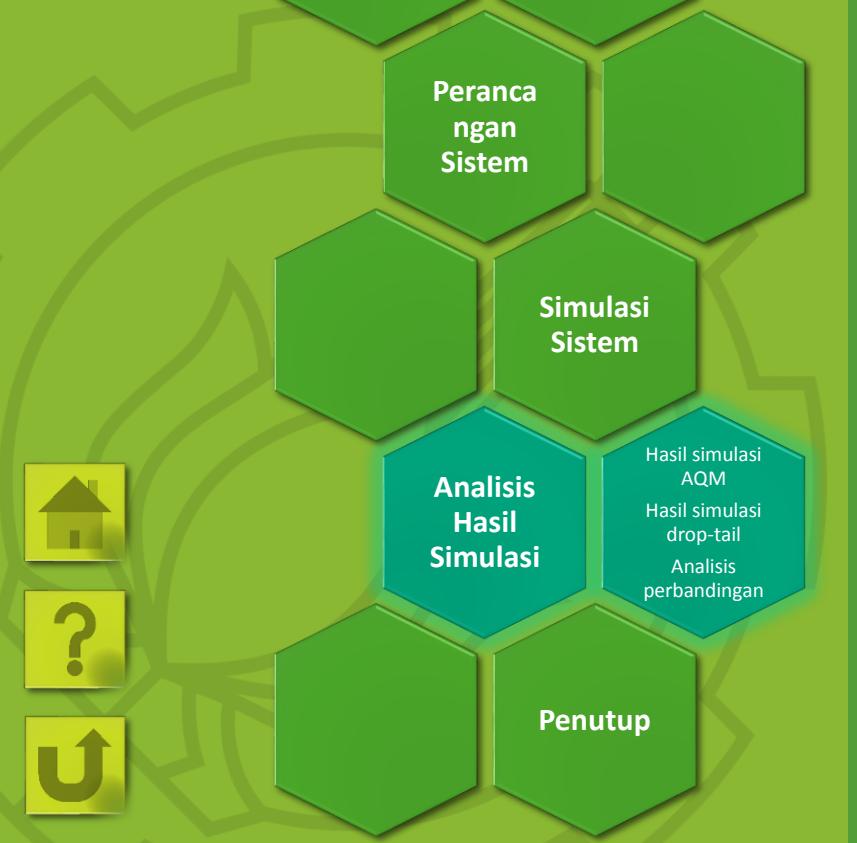
Hasil simulasi AQM

Untuk K = 60

Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	8.87346788	367.671217	65.19304
200	8.3191557	613.271845	66.67624
300	8.68875078	1047.13556	68.81581
400	8.00070601	1558.0204	67.8437
500	11.6272051	2160.2805	66.21581

Untuk K = 108

Volume Requests	Packet-loss (%)	Total Delay (ms)	Jitter (ms)
100	11.5380262	376.386611	65.62361
200	10.753427	613.271845	66.67624
300	11.2180261	1024.22901	69.02137
400	9.73555488	1558.35719	67.8449
500	9.56201288	1787.71267	68.08448



# Analisis Hasil Simulasi

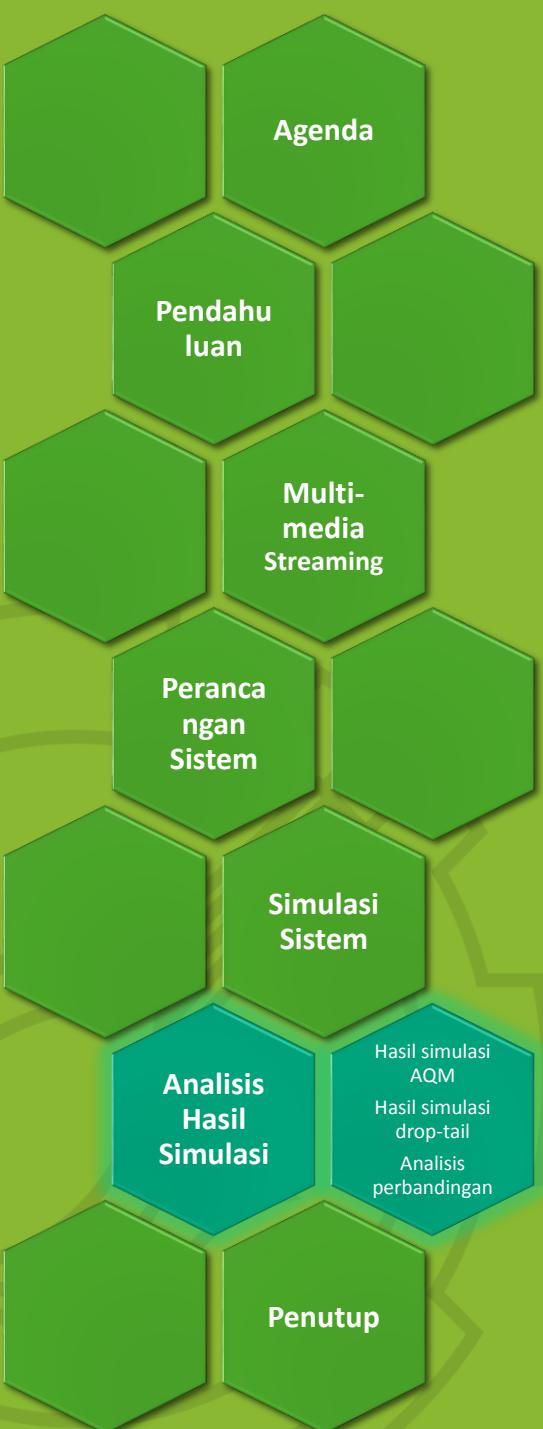
Hasil simulasi drop-tail

Untuk 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

Untuk K = 108

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

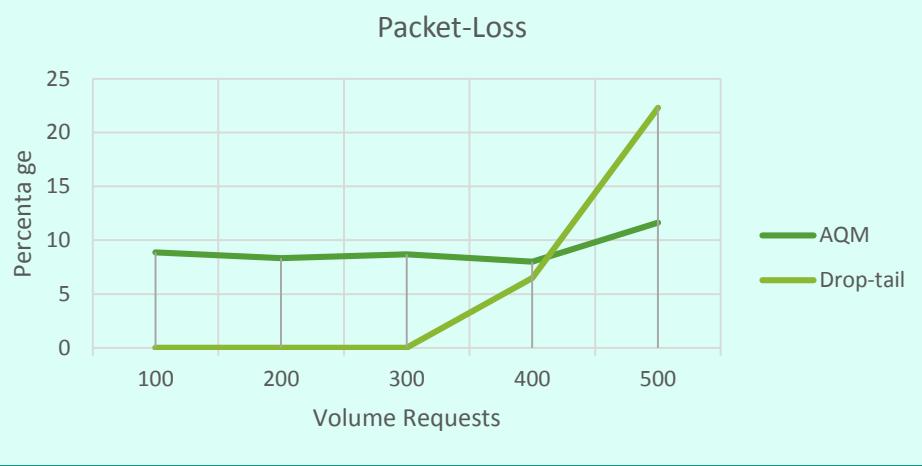


# Analisis Hasil Simulasi

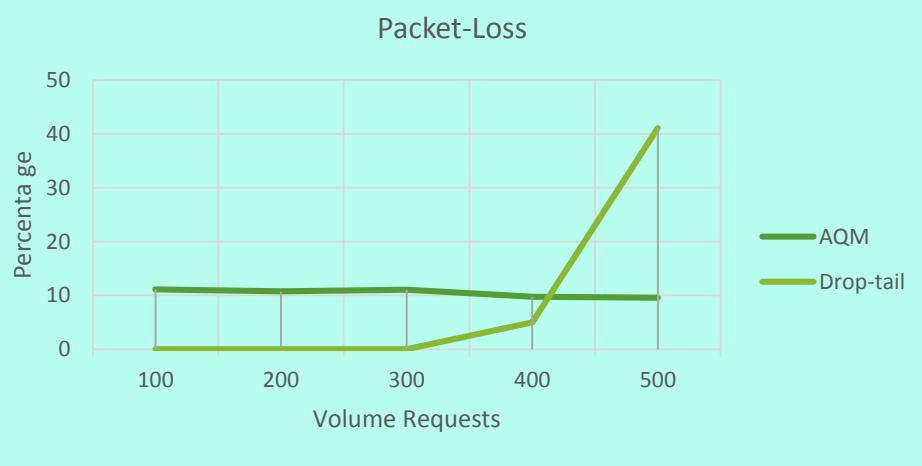
Analisa perbandingan

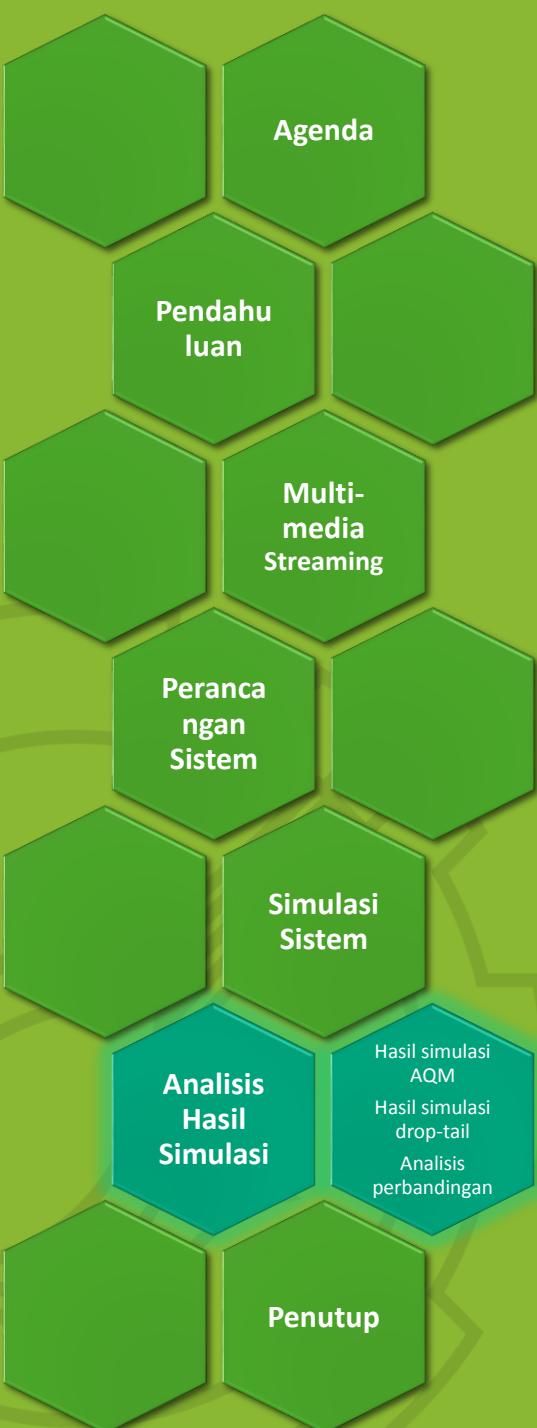
## Packet-loss

Untuk K = 60



Untuk K = 108





# Analisis Hasil Simulasi

Analisa perbandingan

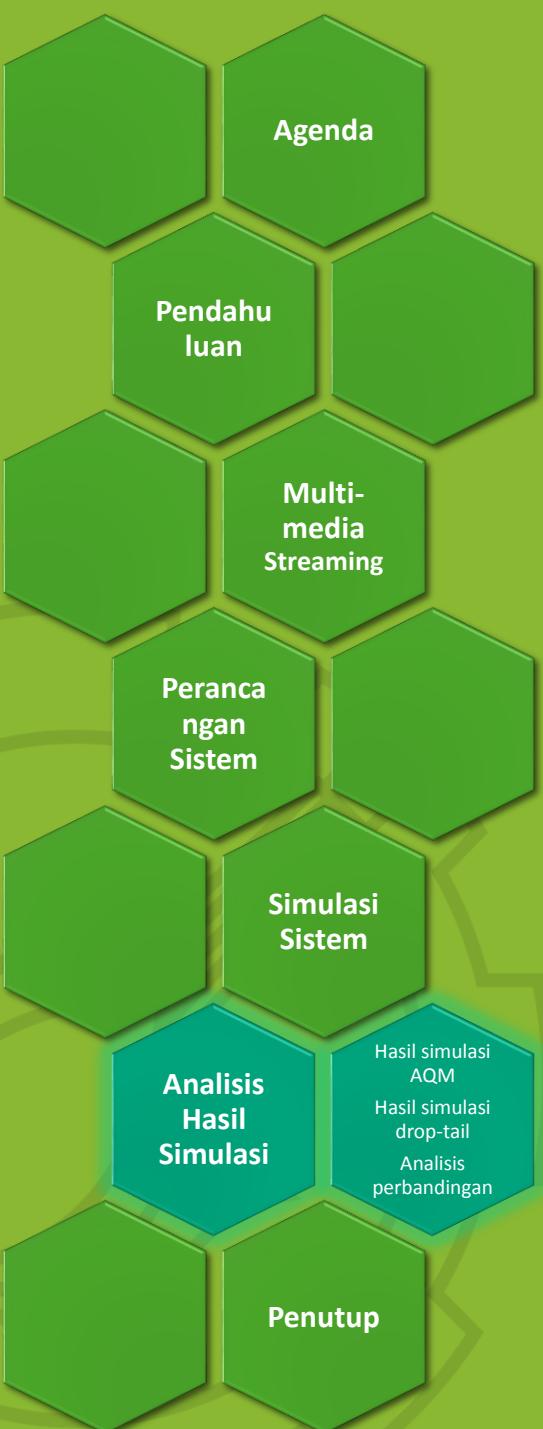
## Total delay

Untuk K = 60



Untuk K = 108



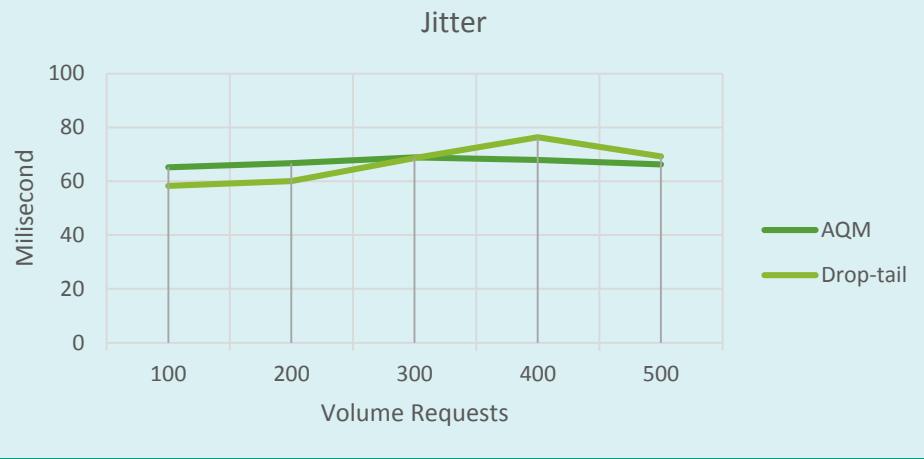


# Analisis Hasil Simulasi

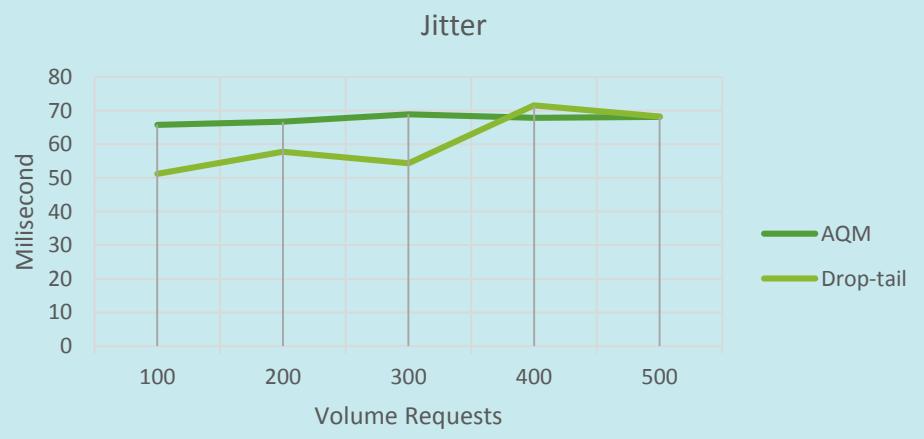
Analisa perbandingan

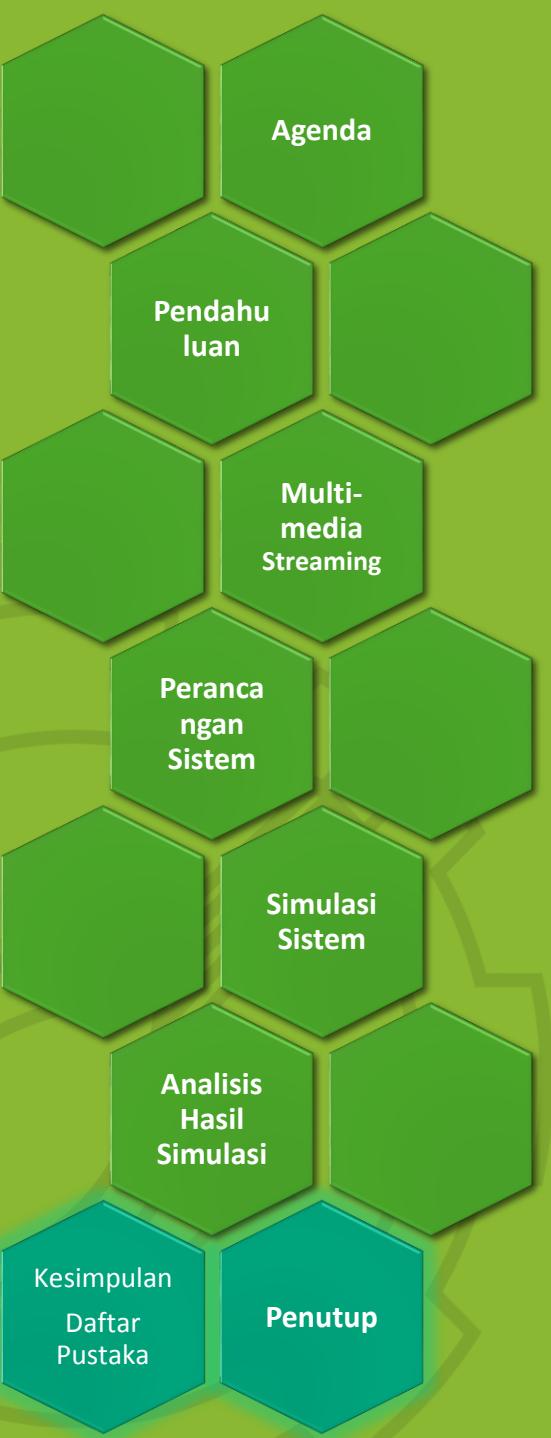
## Jitter

Untuk K = 60



Untuk K = 108

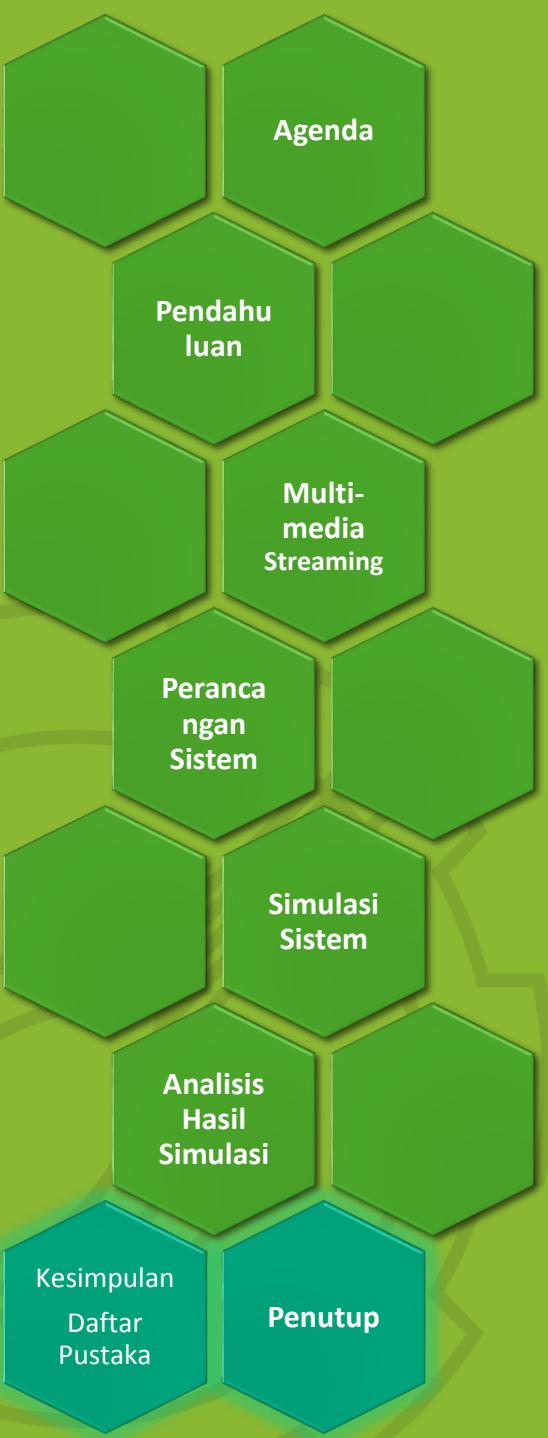




# Penutup

## Kesimpulan

1. Performa sistem antrian pada simulasi sistem antrian dengan algoritma AQM berbasis antrian tergolong sedang yang cenderung baik.
2. Performa sistem antrian pada simulasi sistem antrian dengan algoritma drop-tail tergolong sedang yang cenderung buruk.
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.



# Penutup

## 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] D. Gross, J. F. Shortie, J. M. Thompson and C. M. Harris, Fundamentals of Queueing Theory 4th Edition, United States of America: John Wiley & Sons, Inc, 2008.
- [4] 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.
- [5] M. Mathew, "Overview of Temporal Scalability With Scalable Video Coding (SVC)," Texas Instruments Inc, Texas, 2010.
- [6] Y. G. A. G. N. M. Neda Beheshti, "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. G. S. H. Omar Almomani, "Performance Study of Large Block FEC with Drop Tail for Video Streaming over the Internet," in 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)," in General aspects of Quality of Service (QoS), 1999.
- [11] B. Santosa, "Manajemen Bandwidth Internet dan Intranet," 2004.

Thank you  
for  
listening!



Oleh Ilham Fahmi Kurniawan



# Diskusi

---



Oleh Ilham Fahmi Kurniawan