

## Perbandingan Mean Opinion Score (MOS) pada Jaringan VoIP Menggunakan Proportional Integral Controller Enhanced (PIE) dan Droptail

Fadli Rianda<sup>1</sup>, Aji Gautama Putrada Satwiko<sup>2</sup>, Siti Amatullah Karimah<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom Bandung

[Fadli.rianda@gmail.com](mailto:Fadli.rianda@gmail.com)<sup>1</sup>, [ajigps@telkomuniversity.ac.id](mailto:ajigps@telkomuniversity.ac.id)<sup>2</sup>, [siti.a.karimah@gmail.com](mailto:siti.a.karimah@gmail.com)<sup>3</sup>

---

### Abstrak

Voice over Internet Protocol (VoIP) adalah teknologi yang menggunakan internet untuk melakukan komunikasi jarak jauh. VoIP berkembang seiring dengan berkembangnya teknologi *smartphone* yang pesat. Melakukan komunikasi menggunakan telepon konvensional sudah mulai ditinggalkan. Perkembangan teknologi VoIP yang pesat dengan penggunaannya yang sudah sangat banyak mengakibatkan kemungkinan terjadinya *bufferbloat* pada jaringan. Permasalahan ini dapat diatasi dengan mekanisme pengatur antrian atau biasa disebut *Active Queue Management (AQM)*. Ada berbagai macam AQM yang sudah dikembangkan seperti *Proportional Integral Controller Enhanced (PIE)*, *Controlling Delay (Codel)*, *DropTail* dan sebagainya. Pada penelitian ini diimplementasikan dan dianalisis kualitas layanan VoIP dengan menerapkan *Proportional Integral Controller Enhanced (PIE)* dan *DropTail*. Pengukuran kualitas dinilai dengan menggunakan metode perhitungan *Mean Opinion Score (MOS)*. Penilaian terbagi menjadi dua yaitu secara subjektif dan objektif. Secara subjektif, nilai MOS didapatkan dengan mendengarkan langsung kualitas suara. Secara Objektif, nilai MOS didapatkan dengan perhitungan *R-Factor*. Hasil pengujian perbandingan nilai *Mean Opinion Score (MOS)* baik secara subjektif maupun objektif menunjukkan bahwa kualitas VoIP dengan algoritma *Proportional Integral Controller Enhanced (PIE)* lebih baik daripada *DropTail*. Pada parameter *throughput*, *packet loss*, maupun *delay* PIE lebih juga lebih baik daripada *Droptail*. Ini mengindikasikan implementasi algoritma PIE terhadap layanan VoIP lebih baik daripada *Droptail*

**Kata kunci :** VoIP, MOS, AQM, *Proportional Integral Controller Enhanced*, *Droptail*

---

### Abstract

Voice over Internet Protocol (VoIP) is a technology that that allows you to make voice calls using broadband internet connection instead of regular phone line. VoIP growing along with the rapid development of *smartphone* technology. Communication using a conventional (analog) telephone has begun to be abandoned. The rapid development of VoIP technology with its users has resulted in the possibility of *bufferbloat* on the network. This problem can be overcome with a queue regulating mechanism or commonly called *Active Queue Management (AQM)*. There are various types of AQM that have been developed such as *Proportional Integral Controller Enhanced (PIE)*, *Controlling Delay (Codel)*, and etc. In this research, the quality of VoIP that applying the *Proportional Integral Controller Enhanced* and *Droptail*. Quality measurement will use *Mean Opinion Score (Subjective)* and calculation of *R-Factor (Objective)*. The result of the comparison of the *Mean Opinion Score (MOS)* values both subjectively and objectively indicate that VoIP quality with *Proportional Integral Controller Enhanced (PIE)* algorithm is better than *Droptail*. The parameters of *throughput*, *packet loss*, and *delay* show that PIE also better than *Droptail*. This indicates that the implementation of the PIE algorithm for VoIP services is better than *Droptail*.

**Keywords :** VoIP, MOS, AQM, *Proportional Integral Controller Enhanced*, *Droptail*

---

## 1. Pendahuluan

### Latar Belakang

Semakin cepatnya perkembangan zaman seperti saat sekarang ini, masyarakat banyak dibantu dengan teknologi untuk melakukan kegiatan sehari-hari. Teknologi ini rata-rata menggunakan Internet agar dapat berjalan dengan semestinya. Makin banyaknya pengguna Internet ini mengakibatkan kinerja Internet semakin memburuk. Tingginya permintaan *bandwidth* tidak disertai dengan kualitas *latency* yang bagus. Paket data yang dirancang untuk mengalir secepat mungkin membuat *buffer* cepat penuh. *Buffer* diperlukan untuk menampung sementara arus paket yang cepat. *Buffer* yang terus dialiri arus paket lama-kelamaan menjadi penuh dan mengakibatkan waktu penanganan paket menjadi sangat lama. Jika hal ini terjadi terus-menerus, akan menyebabkan *latency* yang tinggi atau biasanya fenomena ini disebut *bufferbloat*. Fenomena *bufferbloat* bisa terjadi dimana saja. *Bufferbloat* merupakan salah satu kondisi buffer dengan ukuran besar dan cenderung selalu penuh dan

menyebabkan antrian panjang di dalam buffer, jika hal ini terjadi terus-menerus maka akan menyebabkan jeda transmisi yang tinggi [4].

Solusi dari fenomena *bufferbloat* adalah dengan menggunakan *Active Queue Management (AQM)* yang berguna untuk memantau dan mengelola *queue*. Pada *router* kebanyakan menggunakan metode *first in-first out (FIFO)* yang biasanya disebut *Droptail*. Ada banyak macam *Active Queue Management (AQM)*, seperti *CoDel*, *PIE*, *RED*, *HRED*, dsb. *Proportional Integral Controller Enhanced (PIE)* dipercaya bagus dalam menangani koneksi yang *real-time*.

*Voice over Internet protocol (VoIP)* adalah salah satu teknologi yang bersifat *real-time*. Teknologi ini memungkinkan pengguna untuk melakukan panggilan telepon melalui jaringan IP [2]. Aplikasi VoIP ini bersifat *realtime* membutuhkan *jitter* dan *delay* yang rendah. Pada saat ini penyedia VoIP di internet lebih memilih protokol UDP ketimbang TCP sebagai protokol *transport*. Pada UDP tidak adanya *congestion control* mengakibatkan *delay* dan *jitter* pada *receiver* menjadi besar. Hal ini menyebabkan masalah pada jaringan [3].

Penggunaan VoIP memiliki banyak manfaat seperti komunikasi lintas negara. Teknologi ini berbeda dengan komunikasi konvensional yang memakan banyak biaya karena dengan menggunakan VoIP cukup dengan menyediakan software VoIP decoder dan akses layanan internet untuk melakukan komunikasi melalui jaringan IP.

*Proportional Integral Controller Enhanced (PIE)* merupakan salah satu algoritma *Active Queue Management (AQM)* dengan tujuan utama desainnya adalah kontrol *latency* yang rendah, penggunaan *link* yang tinggi, implementasi yang mudah, kestabilan terjamin dan respon yang cepat [5]. Pada tugas akhir ini akan meninjau kinerja PIE dalam mengatasi jaringan yang sifatnya *real-time* dengan meimplementasikannya pada jaringan VoIP.

### Topik dan Batasannya

Berdasarkan latar belakang di atas terdapat empat node pada topologi sistem yaitu *client1*, *client2*, *router*, dan *server* dengan layanan VoIP menggunakan Asterisk. Bandwidth pada router di kurangi agar terjadi *bottleneck* yang lama kelamaan mengakibatkan *bufferbloat*, hal ini akan mengakibatkan turunnya kualitas layanan VoIP. Pada penelitian ini dengan menerapkan algoritma *Proportional Integral Controller Enhanced* untuk mengurangi dampak *bufferbloat* pada kualitas layanan VoIP dengan melihat dari nilai *Mean Opinion Score (MOS)* secara objektif dan subjektif.

### Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah untuk menganalisa kualitas layanan VoIP yang menerapkan algoritma *Proportional Integral Controller Enhanced (PIE)* dan *Droptail*.

### Organisasi Tulisan

Penulisan tugas akhir ini disusun dalam beberapa bagian yaitu Bagian 1 – pendahuluan berisi pendahuluan untuk menjelaskan hal yang dilakukan pada penelitian ini, Bagian 2 – Studi Terkait berisi tentang jurnal atau penelitian yang berkaitan dengan penelitian yang sedang dilakukan kali ini sehingga bisa dijadikan referensi untuk penelitian ini, Bagian 3 – Sistem yang dibangun berisi penjelasan tentang spesifikasi dan alur kerja pada sistem yang dibangun, Bagian 4 – Evaluasi berisi tentang analisis dan evaluasi dari hasil pengujian yang telah dilakukan, Bagian 5 – Kesimpulan yang didapatkan dari hasil penelitian dan saran membangun untuk penelitian kedepannya.

## 2. Studi Terkait

### 2.1. Voice over Internet Protocol (VoIP)

Voice over Internet Protocol (VoIP) adalah integrasi dari layanan telepon konvensional dengan semakin bertambahnya jumlah aplikasi berbasis IP (*IP-based applications*), dan teknologi ini adalah salah satu teknologi yang penting untuk penyedia layanan telekomunikasi [12]. Layanan ini lebih memungkinkan membuat panggilan telepon melalui koneksi internet.

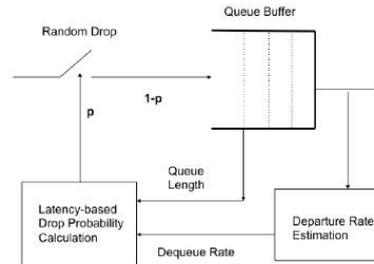
Cara kerja VoIP yaitu dengan pengiriman sebuah sinyal secara digital. Sebelum proses pengiriman (transmisi) dilakukan, data yang berupa sinyal analog akan dikonversikan terlebih dahulu dengan ADC (*Analog to Digital Converter*) menjadi bentuk data digital. Setelah proses ini dilakukan data digital akan di transmisikan ke sumber tujuan. Setelah sampai, data sinyal digital akan dikonversikan kembali menjadi data analog DAC (*Digital to Analog Converter*) sehingga dapat diterima oleh sumber tujuan dengan data sinyal yang ditransmisikan [13].

### 2.2. Active Queue Management (AQM)

Active Queue Management adalah suatu metode dimana perangkat jaringan dapat mengontrol panjang antrian atau lama waktu paket yang diperbolehkan untuk tetap berada di buffer pada perangkat. *Active Queue Management (AQM)* merupakan salah satu cara untuk menangani kondisi *buffer* dengan ukuran besar yang cenderung selalu penuh dan menyebabkan antrian panjang di dalam *buffer* yang bias menyebabkan jeda transmisi yang tinggi jika terjadi terus-menerus atau biasa disebut *bufferbloat* [2].

### 2.3. *Proportional Integral Controller Enhanced (PIE)*

Pada tahun 2013, algoritma *Proportional Integral Controller Enhanced* mulai dikenalkan. PIE sendiri adalah gabungan dari manfaat RED dan CoDel : mudah diimplementasikan seperti RED dan tetap mengontrol *latency* seperti CoDel[10].



Gambar 1- Skema PIE

#### a) *Random Dropping*

Seperti pada kebanyakan AQM, PIE akan men-*drop* paket secara *random* tergantung pada *drop probability* ( $p$ ), dimana  $p$  didapatkan dari komponen *drop probability calculation*.

#### b) *Drop Probability Calculation*

Berikut adalah *drop probability* pada algoritma PIE :

- Estimasi *delay* pada *queue*

$$cur_{del} = \frac{qlen}{avg_{drate}}$$

- Menghitung nilai  $p$

$$p = p + a * (cur_{del} - ref_{del}) + \beta * (cur_{del} - old_{del})$$

- *Update delay*

$$old_{del} = cur_{del}$$

#### c) *Departure Rate Estimation*

Pada saat paket dikirim :

- Memutuskan untuk berada dalam siklus pengukuran, jika :

$$qlen > dq_{treshold};$$

- Apabila nilai di atas *True*, perbarui hitungan keberangkatan  $dq_{count}$  :

$$dq_{count} = dq_{count} + dq_{pktzise};$$

- Perbarui *departure rate*  $dq_{count} > dq_{treshold}$  dan reset penghitungan ;

$$dq_{int} = now - start;$$

$$dq_{rate} = \frac{dq_{count}}{dq_{int}};$$

$$avg_{drate} = (1 - \epsilon) * avg_{drate} + \epsilon * dq_{rate};$$

$$start = now;$$

$$dq_{count} = 0$$

#### d) *Handling Bursts*

Tiga komponen di atas adalah komponen dasar dari algoritma PIE. Berikut ini adalah perhitungan *burst allowance* dari algoritma PIE,

Pada saat paket diterima :

- If  $burst\_allow > 0$

*Enque packet bypassing random drop;*

Pada saat  $dq_{rate}$  diperbarui

- *Update burst allowance:*

- If  $p = 0$ ; and both  $cur_{del}$  and  $old_{del}$  less than  $ref_{del}/2$ , reset  $burst_{allow}$ ,

$$burst_{allow} = max\_burst;$$

Jadi semua komponen di atas adalah komponen penting pada skema algoritma PIE untuk mengontrol *latency* tetap rendah dan untuk menjaga tidak terjadinya *bufferbloat*. Pada Linux (dengan kernel 3.5 atau di atasnya), modul kernel PIE dapat diterapkan menggunakan *tc* atau *traffic control*.

## 2.4. Droptail

Drop Tail adalah algoritma AQM yang sangat sederhana yang sudah dipasang di banyak *router*. AQM ini tidak membedakan *traffic* dari sumber yang berbeda. Selama antrian terisi, paket berikutnya akan di *drop*. Dengan kata lain, AQM ini akan *drop* ekor (*tail*) dari urutan paket.

Ide dari AQM Drop Tail ini adalah paket pertama yang tiba di *router* adalah paket pertama yang akan dikirimkan. Mengingat bahwa jumlah ruang *buffer* yang terbatas pada setiap *router*, jika sebuah paket tiba dan antrian penuh, maka *router* akan membuang paket tersebut. Hal ini dilakukan tanpa mempedulikan aliran apa paket tersebut atau berapa pentingnya paket tersebut. Kinerja sistem antrian M/M/1 tergantung pada parameter berikut:

- Tingkat kedatangan paket.
- Ukuran paket.
- Kapasitas layanan.

Antrian M/M/1 umumnya digambarkan oleh proses Poisson yang mengatur kedatangan paket ke *buffer* terbatas. Ketika sebuah paket mencapai kepala *buffer*, diproses oleh server dan dikirim ke tujuannya [14].

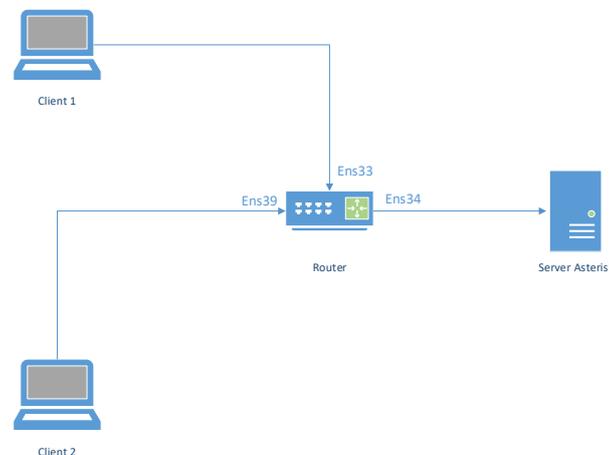
## 2.5. Contoh kasus

Pada tahun 2017, Muhammad Noer Iskandar melakukan penelitian berjudul “Active Queue Management (AQM) Performance Analysis Based On Controlled Delay (CoDel) Algorithm On User Datagram Protocol (UDP)”[11]. Pada penelitian ini melakukan studi analisis terhadap performansi algoritma CoDel, RED, dan Drop Tail pada protocol UDP. Pada riset ini menjelaskan bahwa performansi algoritma CoDel lebih unggul 90% dalam menjaga latency tetap rendah dibandingkan algoritma RED dan Drop Tail. Drop Tail dan RED masi unggul dalam 20% dalam menjaga *throughput* dan nilai *packet-loss* yang ditunjukkan 10% lebih kecil daripada algoritma CoDel.

## 3. Sistem yang Dibangun

### 3.1. Topologi Jaringan

Penelitian ini membangun sebuah sistem yang dapat menghitung nilai *Mean Opinion Score* (MOS) dan *R-Factor* pada jaringan VoIP. Sistem ini menggunakan VMware Workstation dengan beberapa node yaitu *client 1*, *client 2*, *router*, dan *server Asterisk* dimana *client 1*, *router*, *server Asterisk* berada pada 1 *Personal Computer* (PC) dan *client 2* pada Laptop.



Gambar 2- Topologi

Pada *interface router* untuk *client 1* dan *client 2*, *bandwidth* akan disesuaikan sampai mendapatkan angka yang menunjukkan perbedaan yang jelas antara menggunakan AQM PIE dan Droptail. *Active Queue Management* sendiri diatur pada *router*. Pada linux, modul kernel PIE dapat diterapkan menggunakan *tc* atau *traffic control*.

#### a) Client

*Client* berupa komputer virtual yang menggunakan *operating system* Linux yang di *install software softphone* Zoiper.

#### b) Server Asterisk

Asterisk diimplementasikan pada secara *virtual* menggunakan *operating system* Linux.

#### c) Router

*Router* menggunakan Linux pada sebuah komputer *virtual*.

### 3.2. Skenario Perancangan Uji

Skenario uji pada penelitian ini *router* menghubungkan antar *client* dan *server* Asterisk. Pengujian dilakukan dengan mengecilkan *bandwidth* pada *interface router* untuk *client 1* dan *client 2* sehingga terjadi *bottleneck*.

a) **Kondisi Bottleneck 90 Kbps dan 90Kbps**

Skenario ini dilakukan pengecilan *bandwidth* pada *interface router* dengan kecepatan 90Kbps yang kemudian *client 1* melakukan VoIP *call* ke *client 2*. Komunikasi dilakukan dengan menambahkan algoritma Droptail dan PIE.

b) **Kondisi Bottleneck 80 Kbps dan 80Kbps**

Skenario ini dilakukan pengecilan *bandwidth* pada *interface router* dengan kecepatan 80Kbps yang kemudian *client 1* melakukan VoIP *call* ke *client 2*. Komunikasi dilakukan dengan menambahkan algoritma Droptail dan PIE.

c) **Kondisi Bottleneck 70Kbps dan 70Kbps**

Skenario ini dilakukan pengecilan *bandwidth* pada *interface router* dengan kecepatan 70Kbps yang kemudian *client 1* melakukan VoIP *call* ke *client 2*. Komunikasi dilakukan dengan menambahkan algoritma Droptail dan PIE.

### 3.3. Parameter Uji

Terdapat tiga parameter uji yang digunakan dalam menganalisa mekanisme Droptail dan PIE.

a) **Delay**

*Delay* merupakan waktu tunda dalam suatu pemrosesan data, dimana untuk kualitas *delay* dikatakan baik apabila waktu tundanya hanya sekitar 0 – 150 ms [7]. Rata-rata *delay* dapat dihitung seperti persamaan(1).

$$\text{rata - rata delay} = \frac{\text{Total delay}}{\text{Total paket yang diterima}} \dots(\text{persamaan 1})$$

b) **Throughput**

*Throughput* adalah jumlah total *byte* data yang diterima oleh *receiver* pada selang waktu saat paket pertama dan terakhir dikirimkan. Persamaan *throughput* adalah sebagai berikut (2).

$$\text{Throughput} = \frac{\text{Jumlah data yang dikirimkan}}{\text{Waktu pengiriman data}} \dots(\text{persamaan 2})$$

c) **Packet Loss**

*Packet Loss* adalah jumlah paket yang hilang dalam suatu pengiriman paket data pada suatu jaringan. Yang diukur pada penelitian ini adalah perbandingan jumlah paket yang dikirimkan dengan jumlah paket yang dibuang dalam jaringan selama waktu pengujian. Berikut adalah persamaannya(3).

$$\text{Packetloss} = \frac{(\text{Packet}_{\text{transmitted}} - \text{Packet}_{\text{Received}})}{\text{Packet}_{\text{transmitted}}} \times 100\% \dots(\text{persamaan 3})$$

d) **R-Factor**

*R-Factor* adalah salah satu cara untuk mengukur nilai MOS kualitas VoIP dalam jaringan IP secara objektif. Nilai ini berasal dari beberapa parameter seperti *delay* dan gangguan jaringan. *R-Factor* berkisar antara 0 (kualitas sangat buruk) sampai 100 (kualitas tinggi). Setiap *R-Factor* yang nilainya kurang dari 50 tidak dapat diterima. TDM berbasis telepon memiliki *R-Factor* 94. Ada 3 variasi utama *R-factor* yaitu *R-Call Quality Estimate*, *R-Listening Quality Estimate* dan *R-Network Performance Estimate* [8]. Berikut dijabarkan rumus bagaimana cara memperoleh MOS secara objektif [9](4)(5),

$$R = 94.2 - I_d - I_{ef} \dots(\text{persamaan 4})$$

Yang mana :

$I_d$  = faktor dari penurunan kualitas suara oleh suatu *delay*.

$I_{ef}$  = faktor dari penurunan kualitas karena teknik kompresi dan *packet loss*.

$$I_d = 0,024d + 0,11(d - 177,3) \times H_{(d-177,3)}$$

$$I_{ef} = 7 + 30\ln(1 + 15e)$$

Yang mana :

$R$  = faktor kualitas pengiriman

$d$  = *delay*

$H$  = fungsi *heavyside*, yang mana:

$$H_{(x)} = 0, \text{ if } x < 0, \text{ dan}$$

$$H_{(x)} = 1, \text{ if } x \geq 0$$

$e$  = persentase terjadi *packet loss*

Konversi nilai  $R$  ke MOS, berikut aturan yang harus digunakan:

- $R < 0$  : MOS = 1,
- $R > 100$ : MOS = 4,5,
- $0 < R < 100$ ;  $MOS = 1 + 0,035R + (7 \times 10^{-6})R(R - 60)(100 - R)$

e) **Nilai Mean Opinion Score (MOS)**

*Mean Opinion Score* (MOS) adalah sebuah penilaian yang didapat dari mendengarkan sebuah suara secara langsung, sistem penilaian ini bersifat subjektif. Berikut adalah tabel nilai MOS.

No	Kualitas	Nilai
1	Sangat Baik	5
2	Baik	4
3	Cukup	3
4	Buruk	2
5	Sangat Buruk	1

Tabel 1- Nilai Kualitas MOS

#### 4. Evaluasi

##### 4.1. Peformansi Droptail

Traffic Shaping (Kbps)	Rata-Rata Delay (ms)	Throughput (Mbit/s)	Dropped Packet		
			Receive	Expected	Loss
90	19.986686	0,168	11020	11020	0%
80	41.797767	0,112	8157	10996	25.82%
70	99.270147	0.08	4582	9297	50,72%

Tabel 2- Peformansi Droptail

Pada tabel 2 menjelaskan peformansi Droptail pada setiap skenario *bottleneck* yaitu 90 Kbps, 80 Kbps, 70 Kbps. Pada tabel 2 menjelaskan semakin kecil nilai *traffic shaping* maka semakin besar nilai rata-rata *delay* dan *packet loss*, sedangkan nilai *throughput* semakin kecil.

##### 4.2. Peformansi Proportional Integral Controller Enhanced (PIE)

Traffic shaping (Kbps)	Rata-Rata Delay (ms)	Throughput (Mbit/s)	Dropped Packet		
			Receive	Expected	Loss
90	19.994444	0,168	11017	11017	0%
80	25.927081	0,128	9529	10761	11.45%
70	62.84687	0,096	4093	6786	39.68%

Tabel 3 - Peformansi Proportional Integral Controller Enhanced

Pada tabel 3 menjelaskan peformansi Proportional Integral Controller Enhanced (PIE) pada setiap skenario *bottleneck* yang diujikan. Tabel tersebut juga menjelaskan semakin kecil nilai skenario maka semakin besar nilai rata-rata *delay* dan *packet loss*, sedangkan nilai *throughput* semakin kecil.

##### 4.3. Nilai Mean Opinion Score (MOS)

Pada tabel 4 menjelaskan tentang kualitas suara secara subjektif. Pada penelitian ini terdapat 30 orang yang mendengarkan rekaman dari semua skenario yang diujikan, lalu mengisi nilai antara 1 – 5. Pada skenario 1 nilai MOS tidak terlalu signifikan, tetapi pada skenario 2 dan 3 mulai terlihat nilai yang cukup signifikan.

No	PIE			Droptail		
	90 Kbps	80 Kbps	70 Kbps	90 Kbps	80 Kbps	70 Kbps
1	4	3.5	1.5	4	3	1
2	4	3	3	4	2	2
3	5	3	2	5	2	1.5
4	5	3	2	4	2	1
5	4	3	2	4	2	1
6	4	3	2	4	2	2
7	4	3	3	4	3	3
8	5	3.5	2	4	3	1
9	5	3	3	5	2	3
10	5	4	2	4	2	2
11	4	3	2	4	2	1
12	5	3	1	5	3	1
13	5	3	2	5	2	1.5
14	4	3	3	4	2	2
15	4	3	2	3	2	2
16	5	3	2	4	3	2
17	4	3	1	3	2	1
18	3	2	1	3	2	1
19	3.5	2.5	1	3	2	1
20	4	3	2	3.5	2.5	1.5
21	3.5	3	1.5	3	2.5	1
22	3	2	1	3	2	1
23	5	3	2	5	3	2
24	5	3	2	4	2	1
25	5	3	1	5	3	1
26	4	3	2	4	2	2
27	5	4	2	4	2	2
28	4.5	3	1.5	4	3	1
29	5	3	2	5	2	1
30	3	1.5	0.5	3	1	0.5
Mean	4.317	2.967	1.833	3.983	2.267	1.467

Tabel 4- Nilai Mean Opinion Score (MOS)

#### 4.4. Nilai R-Factor

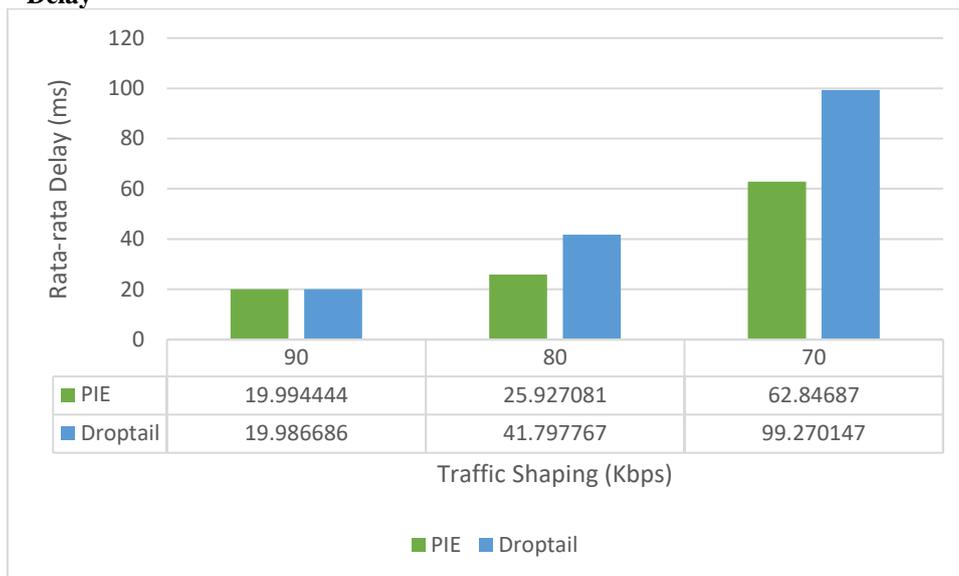
Traffic shaping (Kbps)	R-Factor	
	PIE	Droptail
90	86.720	86.720
80	56.586	38.686
70	27.521	20.237

Tabel 5- Nilai R-Factor

Pada tabel 5 menjelaskan tentang nilai dari R-Factor dari semua skenario yang diujikan. Sama seperti nilai Mean Opinion Score (MOS), perbedaan yang signifikan terlihat pada skenario 2 dan 3.

#### 4.5. Perbandingan Droptail dan PIE

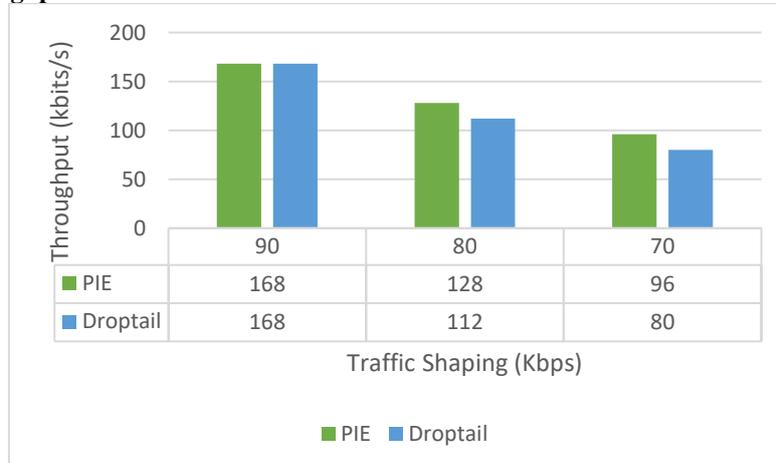
##### a) Delay



Gambar 3- Rata-rata Delay PIE dan,Droptail

Gambar 3 menjelaskan bahwa pada skenario 1 (90 Kbps) perbedaan *delay* tidak terlalu signifikan, akan tetapi pada skenario 2 (80 Kbps) dan skenario 3 (70 Kbps) perbedaan *delay* mulai terasa. Hal ini dikarenakan paket RTP sangat kecil yaitu 85 Kbps sehingga pada skenario 1 tidak terasa perbedaannya. Pada skenario 2 dan skenario 3 algoritma PIE menjaga *delay* agar tetap rendah. Hal ini disebabkan karena algoritma PIE menjaga *latency* tetap rendah dengan cara menggunakan *timestamp* untuk menandai setiap paket dalam antrian. Ketika paket melebihi *timestamp*, maka paket tersebut akan di-*drop* untuk menjaga antrian tidak penuh dan menjaga *delay* tetap rendah.

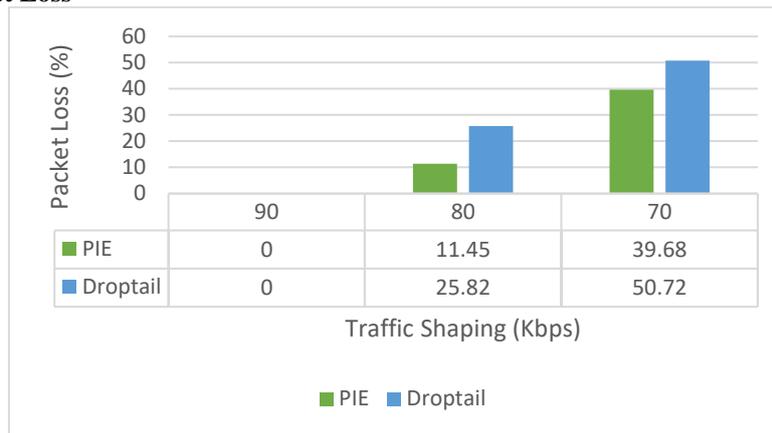
##### b) Throughput



Gambar 4- Throughput PIE dan Droptail

Gambar 4 menjelaskan peformansi kedua algoritma dalam menangani throughput, algoritma PIE lebih besar daripada droptail. Pada setiap skenario memiliki nilai *throughput* yang berbeda, hal ini dikarenakan *bandwidth* yang diberikan. Jadi semakin besar *bandwidth* maka *throughput* juga akan semakin besar. Perbedaan *throughput* antara kedua algoritma PIE dan Droptail tidak terlalu signifikan karena RTP yang sangat kecil. Nilai *throughput* pada PIE lebih bagus karena PIE menjaga *latency* tetap rendah pada *router*.

## c) Packet Loss



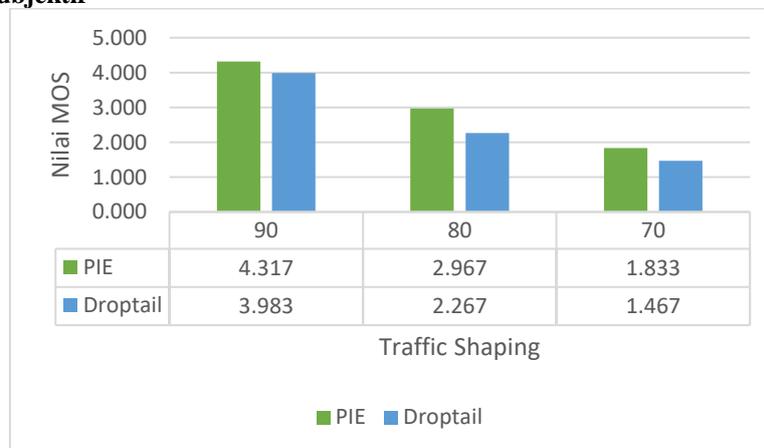
Gambar 5- Packet Loss PIE dan Droptail

Gambar 5 menjelaskan tentang perbandingan *packet loss* pada setiap scenario yang diujikan. Pada skenario 1 tidak terjadi *packet loss* karena kecilnya paket RTP itu sendiri, sehingga paket langsung diantarkan dari *client 1* ke *client 2*. *Packet loss* mulai terjadi pada skenario 2 (80 Kbps) dan skenario 3 (70 Kbps) karena data mulai terjadi *congestion* pada *router* disebabkan *bandwidth* yang disediakan sangat kecil. Algoritma PIE menangani *packetloss* lebih baik karena PIE memiliki hitungan *Drop Probability* dengan menjaga *latency* tetap rendah dan menjaga antrian dalam *router* tetap stabil. sedangkan algoritma Droptail akan membuang (*drop*) semua paket apabila antrian dalam *router* penuh.

## d) Nilai Mean Opinion Score (MOS)

Terdapat dua macam nilai *Mean Opinion Score* (MOS) pada penelitian ini, yaitu secara subjektif dan objektif. Secara subjektif nilai langsung diberikan setelah mendengar semua skenario dimana pada penelitian ini nilai diambil dari rata-rata nilai yang diberikan oleh 30 orang. Secara Objektif nilai MOS diambil dari nilai R-Faktor.

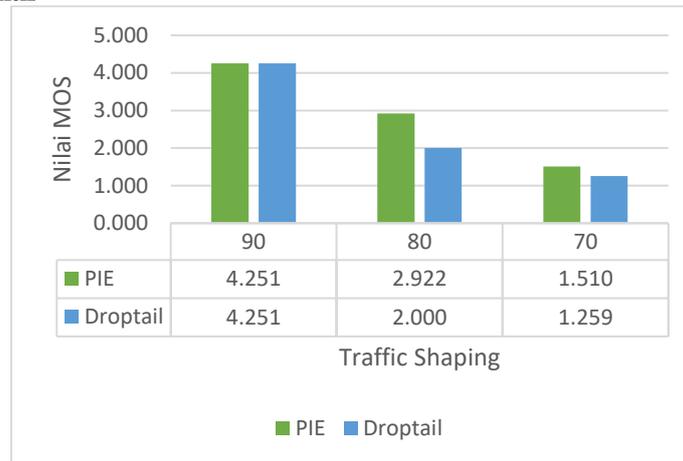
## • Subjektif



Gambar 6- MOS subjektif PIE dan Droptail

Pada gambar 6 terlihat jelas perbedaan nilai MOS antara PIE dan Droptail. Hasil ini diambil dari rata-rata total nilai dari penilaian dari 30 orang yang mendengarkan langsung rekaman *call VoIP* pada setiap skenario. Algoritma PIE mendapatkan nilai MOS lebih baik daripada Droptail, hal ini juga mengacu dari hasil parameter *delay*, *throughput*, dan juga *packetloss* dimana PIE lebih baik daripada Droptail. Jadi semakin bagus penanganan *throughput*, *delay*, dan *packetloss* maka semakin bagus juga kualitas yang dihasilkan.

### • Objektif



Gambar 7- MOS Objektif PIE dan Droptail

Pada gambar 7 merupakan nilai *Mean Opinion Score* (MOS) yang didapatkan dari nilai *R-factor* yang dapat dilihat pada tabel 5. Dapat dilihat secara objektif nilai PIE juga lebih baik daripada Droptail sesuai dengan nilai subjektif yang didapatkan seperti gambar 5. Nilai yang didapatkan juga tidak jauh berbeda. Hal ini membuktikan bahwa dalam memaksimalkan kualitas layanan VoIP, menggunakan algoritma PIE lebih baik daripada Droptail.

### 5. Kesimpulan

Tujuan penelitian ini adalah untuk menguji kualitas MOS pada algoritma *Proportional Integral Controller Enhanced* (PIE) dan Droptail pada jaringan *Voice over Internet Protocol* (VoIP). Hasil perbandingan antara dua algoritma tersebut telah di sajikan pada sub bab 4.5. Dari hasil tersebut dijelaskan bahwa *delay* pada Droptail lebih tinggi daripada PIE dan *packet loss* yang dihasilkan Droptail juga lebih tinggi daripada PIE. Dari segi *throughput*, PIE lebih unggul dari pada Droptail. Nilai *delay*, *packet-loss* dan *throughput* ini akan mempengaruhi kualitas MOS baik secara subjektif maupun objektif. Nilai MOS pada algoritma PIE lebih bagus dikarenakan sistem *random dropping* yang membuang paket secara acak membuat kualitas suara lebih jelas, sedangkan Droptail akan membuang semua paket apabila antrian sudah penuh.

### 6. Saran

Saran yang bisa diterapkan untuk penelitian ini lebih lanjut agar implementasi algoritma *Active Queue Management* (AQM) ini bisa diimplementasikan langsung pada jaringan 3G atau 4G yang ada di Indonesia dan dianalisis pada saat jaringan itu dalam keadaan padat yang akan menghasilkan *congestion* atau hambatan.

### Daftar Pustaka

1. Z.Nasution. (2014). <http://repository.ut.ac.id>. Retrieved from <http://repository.ut.ac.id>: <http://repository.ut.ac.id/4482/2/SKOM4322-TM.pdf>
2. Satria Mandala, M. N. (2017, Maret 1). Active Queue Management (AQM) Performance Analysis based on Controlled Delay (CoDel) Algorithm on UDP. *Indonesian Journal of Computing*, 2, 1. doi:10
3. La Surimi, R. P. (2015, July). Analisis Kualitas VoIP pada SCTP Menggunakan ECN dan AQM. *Indonesian Journal of Computing and Cybernetics Systems*, 9, 123.
4. Nur Lina, 2014, Congestion (pada jaringan data), (online) <http://nurlinaamik.blogspot.co.id/2014/06/congestion-pada-jaringan-data.html>. Diakses : 10 November 2017.
5. Allied Telesis "QOS White Paper". <https://www.alliedtelesis.com/documents/qos-white-paper>. Diakses : 12 November 2017.
6. Ginanjar Rahmansyah, T. B. (2015). Analisis Mekanisme Active Queue Management (AQM) Berbasis Controlled Delay., II, p. 1145. Bandung. Retrieved 8 14, 2018
7. ITU-T Series G. 2003. "Transmission system and media, Digital systems and networks".
8. Andrew, 2011, "Difference Between R-Factor and MOS Score in VoIP Quality". <http://www.differencebetween.com/difference-between-r-factor-and-mos-score-in-voip-quality/>. Diakses : 12 November 2017.

9. Vouzis, P. (2016, August 18). *Impact of Packet Loss, Jitter, and Latency on VoIP*. Retrieved August 21, 2018, from NetBeez: <https://netbeez.net/blog/impact-of-packet-loss-jitter-and-latency-on-voip/>
10. Rong Pan, P. N. (2013). PIE: A Lightweight Control Scheme to Address the. *2013 IEEE 14th International Conference on High Performance Switching and Routing* (pp. 148-155). Taipei: IEEE. doi:10.1109/HPSR.2013.6602305
11. ISKANDAR, M. N. (2017). *ANALISIS PERFORMANSI ACTIVE QUEUE MANAGEMENT (AQM) BERBASIS ALGORITMA CONTROL DELAY (CoDel) PADA STREAMING USER DATAGRAM PROTOCOL (UDP)*. (S. M. MUH. ARIF NUGROHO, Ed.) Bandung, Indonesia: Telkom.
12. Nobuhiko Kitawaki, A. T. (2004, July 28). Perceptual QoS Assessment. *perceptual QoS assessment technologies for VoIP*, 28 - 34. doi:10.1109/MCOM.2004.1316526
13. Setiawan, E. B. (2012, October). ANALISA QUALITY OF SERVICES (QoS) VOICE OVER INTERNET PROTOCOL (VoIP) DENGAN PROTOKOL H.323 DAN SESSION INITIAL PROTOCOL (SIP). *ANALISA QUALITY OF SERVICES (QoS) VOICE OVER INTERNET PROTOCOL (VoIP) DENGAN PROTOKOL H.323 DAN SESSION INITIAL PROTOCOL (SIP)*, 7 - 8. Retrieved from <https://ojs.unikom.ac.id/>
14. Misbahul, F. (2016, November). Simulasi Antrian Paket Data Jaringan dengan Mekanisme Drop Tail. *Jurnal Ilmiah FIFO, VIII*, 151 - 160. Retrieved 11 10, 2017