

## BAB IV ANALISA DAN HASIL

Hasil yang diperoleh dari pengolahan data dan simulasi ini ada berbagai macam jenis data output yang diinginkan. Seperti *file trace video*, *file trace sender* serta *file trace receiver* yang akan diolah informasinya seperti *throughput*, *jitter*, *paket loss*, *delay* serta *psnr*. Hasil-hasil inilah yang akan dijadikan penghitungan sehingga terdapat perbandingan antara data sebelum simulasi dan setelah simulasi yang dilakukan berdasarkan parameter-parameter yang ditentukan dan juga berdasarkan dari hasil pengamatan.

Tabel 4.1 dibawah ini adalah nilai *tunning* yang didapatkan dari melakukan penelitian untuk mencari titik optimum nilai *threshold* dari tiap nilai *Threshold* berurutan 5, 40, 5 dan 40 dengan *paket loss* yang dihasilkan hanya 15 paket.

Tabel 4.1 Nilai Optimum dari *threshold* yang digunakan

<i>Threshold1</i>	<i>Threshold2</i>	<i>Threshold3</i>	<i>Threshold4</i>	Total Paket LOSS
10	50	5	20	17
10	40	5	15	22
10	30	5	10	29
5	50	5	20	16
<b>5</b>	<b>40</b>	<b>5</b>	<b>15</b>	<b>15</b>
5	30	5	10	24

### 4.1 FILE TRACE INPUT

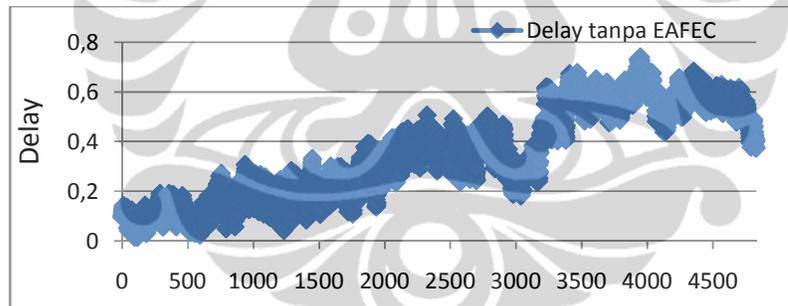
*Video traffic trace* yang digunakan adalah video dengan nama “*highway*” yang menggunakan video *coding* H.264 dengan *tool* JM 1.7 untuk *Codec* nya. Video ini berformat *qcif* 176 x 144. *Video trace* ini terdiri dari 2000 *Frames* dimana masing-masing *Frame* terbagi kedalam *slices* transmisi. Masing-masing *slices* sebesar 500 bytes dan di transmisiikan melalui RTP/UDP/IP paket dengan transmisi bersifat *unicast*. Video ini juga mempunyai *Frame rate* sebesar 30 *Frame/second* dengan total paket video yang dikirimkan sebanyak 4829 paket selama 67 detik.

## 4.2 Throughput

*Throughput* yang merupakan besarnya kecepatan transmisi data yang dikirimkan pada waktu tertentu. *Throughput* di tiap-tiap topologi mempunyai hasil yang berbeda-beda. Pada skenario ke-1 (tanpa mekanisme EAFEC) nilai rata-rata *throughput* sebesar 222,176 kbps dan pada skenario ke-2 didapat nilai rata-ratanya adalah 220,368 kbps. Terdapat selisih antara rata-rata nilai *throughput* nilai yang lebih besar ada pada skenario pertama dikarenakan pada skenario ini tidak terdapat terjadi proses mekanisme EAFEC sehingga keluarannya lebih besar sedangkan pada skenario ke-2 ada proses penambahan mekanisme EAFEC sehingga berpengaruh terhadap transmisi data.

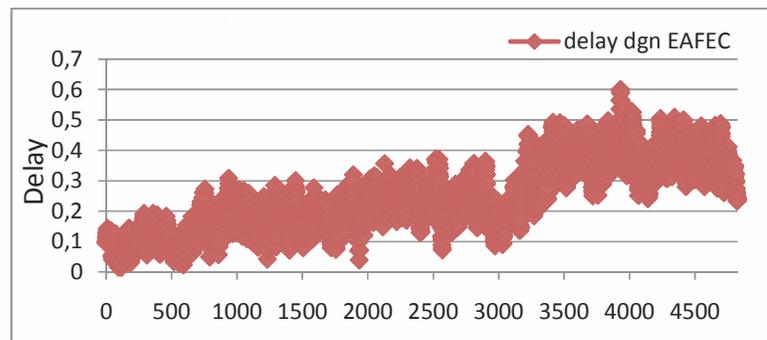
## 4.3 DELAY

*Delay* masing-masing skenario dapat dilihat pada parameter *end-to-end delay*. Dari Gambar 4.1 dapat dilihat bahwa skenario ke-1 (mekanisme FEC statis) didapat nilai rata-rata *delay* sebesar 0,347 detik dan pada skenario ke-2 pada Gambar 4.3 didapat nilai rata-rata *delay* sebesar 0,249 detik sedikit lebih kecil dari nilai rata-rata *delay* pada skenario pertama.



Gambar 4.1. *Delay* mekanisme FEC Statis

Nomor packet



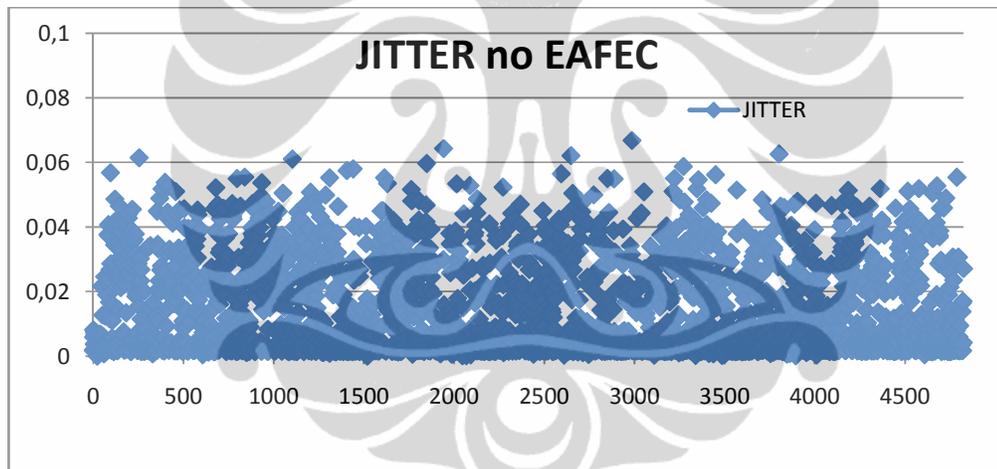
Gambar 4.2 *Delay* dengan mekanisme EAFEC

Nomor packet

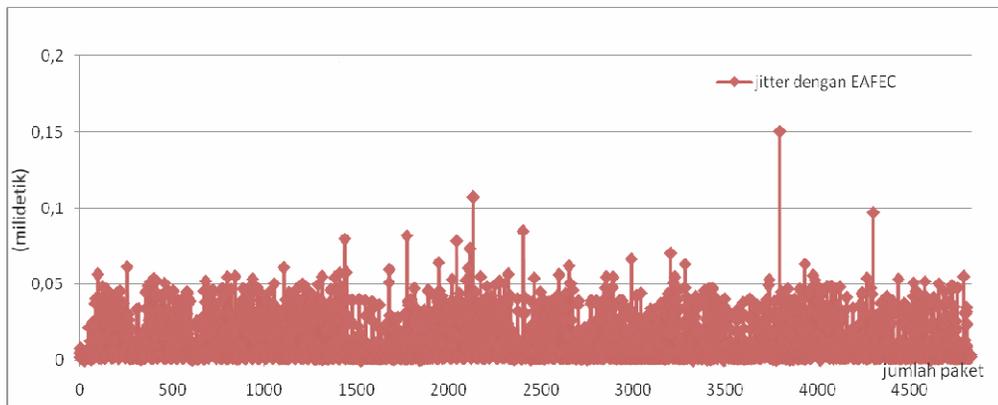
Nilai yang sedikit lebih kecil pada skenario ke-2 dikarenakan paket *drop*-nya lebih kecil dari pada skenario pertama. Disini sistem mengatur antrian untuk mengurangi kehilangan paket akibat jumlah data yang dikirimkan terlalu banyak. Hal ini yang nantinya berpengaruh pada kualitas video yang akan dihasilkan. Dengan kecilnya nilai *delay* maka hasilnya akan lebih berkualitas dari pada skenario pertama yang nilai *delay*-nya lebih besar.

#### 4.4 JITTER

Nilai *jitter* pada pada tiap skenario dijelaskan pada Gambar 4.3 dan Gambar 4.4. Dari hasil pengamatan pada masing-masing gambar didapat dilihat bahwa rata-rata *jitter* dari skenario pertama yaitu mekanisme FEC Statis yaitu 9,670 ms sedangkan pada skenario kedua yang melalui *recovery* EAFEC nilai rata-rata *jitter*-nya yaitu 9,759 ms.



Gambar 4.3 *Jitter* pada mekanisme FEC Statis

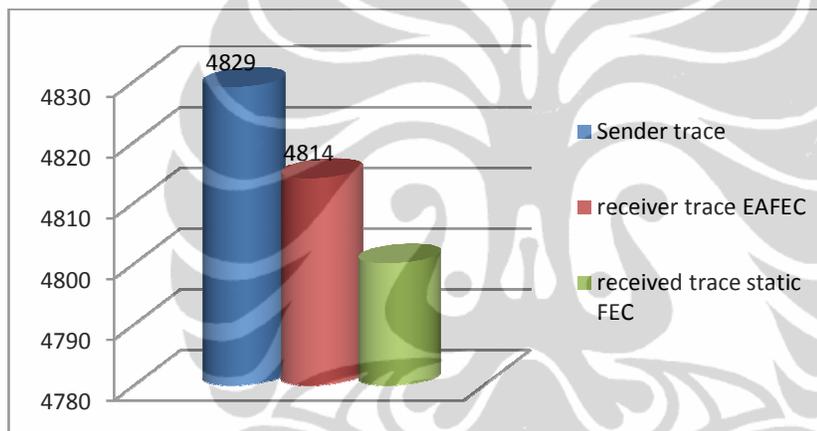


Gambar 4.4 *Jitter* pada skenario dengan EAFEC

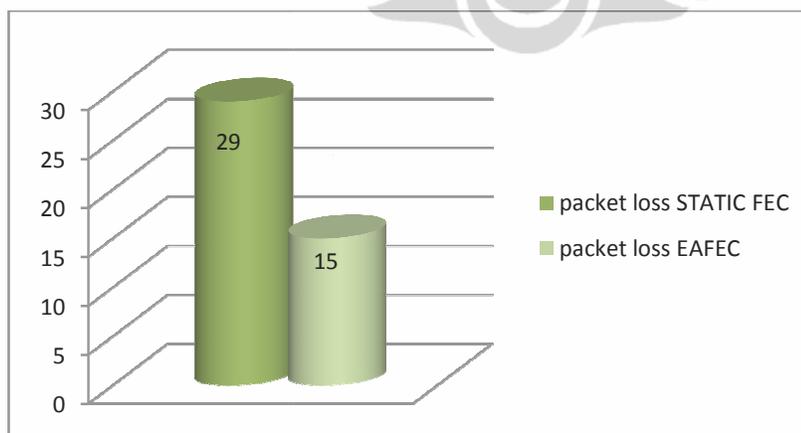
#### 4.5 Packet Loss

Ada perbandingan antara jumlah *packet loss* dari skenario simulasi pertama dengan skenario yang ke dua. Ini terlihat dari hasil persentase masing-masing skenario. Hasilnya dapat dilihat dari visualisasi grafik pada Gambar 4.5 dan Gambar 4.6.

Pada skenario pertama ( FEC Statis ) jumlah *packet loss* sebanyak 29 packet dari total paket yang dikirimkan 4829 paket atau sebesar 0.60 % sedangkan pada gambar 4.6 yaitu skenario dengan perbaikan EAFEC didapat *packet loss* sebesar 15 paket dari total paket yang dikirimkan sebanyak 4829 paket atau sebesar 0,31%.



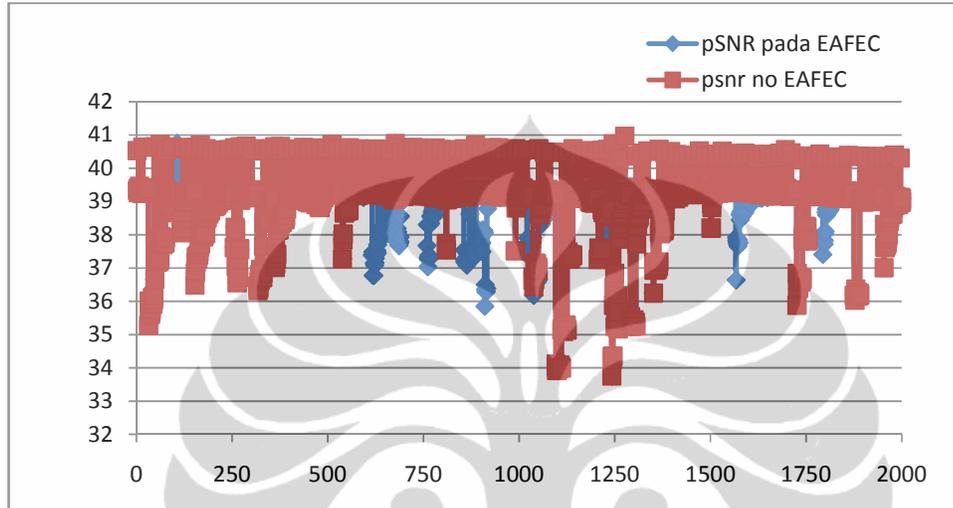
Gambar 4.5 Perbandingan antara jumlah paket yang dikirim dan diterima



Gambar 4.6 Jumlah *Packet loss*

#### 4.6 Peak Signal Noise to Ratio (PSNR)

Pada akhir dari simulasi ini dilakukan pengukuran dari kualitas video yang berdasarkan perbandingan *Frame* dari *file* yang dikirim dan *Frame file* yang diterima. Nilai PSNR juga di visualisasi dengan grafik seperti pada Gambar 4.7 berikut ini.



Gambar 4.7 Perbandingan pSNR pada skenario dengan EAFEC vs Statis FEC.

Dari gambar tersebut didapat nilai rata-rata masing-masing pSNR. Pada skenario pertama yaitu tanpa EAFEC didapat nilai rata-rata dari pSNR sebesar 38,898 dB. Dan pada skenario kedua diperoleh nilai rata-rata pSNR sebesar 38,739 dB. Nilai pSNR ini juga dapat menggambarkan sifat dasar dari karakteristik masing-masing skenario. Perbedaan fluktuasi dari pSNR diatas terjadi karena *packet error/loss*.

#### 4.7 Kualitas Tampilan Video

Pada simulasi ini juga dilakukan perbandingan kualitas dari masing-masing skenario yang didasari dengan nilai subjektif yang diberikan oleh *user*. Didapat 3 jenis video yaitu video mentahan seperti pada tampilan Gambar 4.8a, video static FEC seperti pada Gambar 4.8b dan video dengan optimalisasi algoritma pada Gambar 4.8c. Dari ke-3 jenis video ini didapatkan hasil kualitas terbaik dimiliki oleh video mentahan dikarenakan masih berupa *raw* video yang belum mengalami *distorsi* dan kehilangan jumlah pakatnya dan kualitas

berikutnya ada pada video dengan optimalisasi EAFEC yang ditunjukkan pada Gambar 4.8c. hal ini dikarenakan video yang dihasilkan telah *direcovery* melalui EAFEC sehingga jumlah *packet loss*nya lebih kecil. Dan yang kualitas terakhir adalah video tanpa optimalisasi algoritma. Hal ini dikarenakan jumlah paket lossnya lebih besar dan video yang di-*streaming* tidak mengalami optimalisasi.





Gambar 4.8a Gambar Video Mentahan (*raw video*)



Gambar 4.8b Gambar Video pada Skenario Pertama (FEC Statis)



Gambar 4.8c Gambar Video pada Skenario kedua (dengan EAFEC)