

Analisis Perbandingan Performansi Penjadwalan Paket Antara Homogeneous Algorithm Dengan Hybrid Algorithm Pada Jaringan Point-To-Multipoint Wimax

Dadieq Pranindito¹, Rendy Munadi², R. Rumani M³

¹Sekolah Tinggi Teknologi Telematika Telkom Purwokerto

^{2,3}Telkom University

¹Jl. D. I. Panjaitan No. 128 Purwokerto 53147

^{2,3}Jalam Telekomunikasi Terusan Buah Batu Bandung 40257

dadieqpranindito@gmail.com, rnd@ittelkom.ac.id, rrm@ittelkom.ac.id

Abstrak – Saat ini, dalam dunia telekomunikasi, (*Worldwide Interoperability for Microwave Access*) WiMaX merupakan teknologi nirkabel yang menyediakan hubungan jalur lebar dalam jarak jauh, memiliki kecepatan akses yang tinggi dan jangkauan yang luas serta menyediakan berbagai macam jenis layanan. Masalah yang menarik dan menantang pada WiMaX adalah dalam hal menyediakan jaminan kualitas pelayanan (QoS) untuk jenis layanan yang berbeda dengan bermacam-macam kebutuhan QoS-nya. Untuk memenuhi kebutuhan QoS tersebut, maka diperlukan suatu algoritma penjadwalan.

Dalam penelitian ini dilakukan simulasi jaringan WiMaX menerapkan algoritma penjadwalan dengan metode *homogeneous algorithm* dan *hybrid algorithm*. Perwakilan pada metode *homogeneous algorithm* akan menggunakan algoritma penjadwalan *Weighted Fair Queuing* (WFQ) dan *Deficit Round Robin* (DRR), sedangkan pada metode *hybrid algorithm* menggunakan penggabungan antara algoritma penjadwalan DRR dan WFQ. Pengujian kinerja algoritma penjadwalan tersebut dilakukan dengan membandingkan kedalam 5 jenis kelas QoS pada WiMAX yaitu UGS, rtPS, nrtPS, ertPS, dan *Best Effort*. Dari hasil pengujian, *hybrid algorithm* memberikan nilai QoS yang lebih baik jika dibandingkan dengan *homogeneous algorithm*. *hybrid algorithm* sangat cocok jika diterapkan pada kondisi jaringan yang memiliki trafik dengan paket data yang bervariasi, karena dapat menghasilkan throughput yang tinggi, serta dapat menghasilkan nilai delay dan jitter yang rendah.

Kata kunci - QoS, *homogeneous algorithm*, *hybrid algorithm*.

Abstract – Nowadays, In the telecommunication Words, WiMax (*Worldwide Interoperability for Microwave Access*) is a wireless technology that provides long distance broadband connectivity. It has high speed access, wide coverage areas, and provides various types of services. Applying WiMAX has challenging problem when it has to provide quality assurance of service (QoS) for different types of services with diverse QoS needs.

In this research implement a WiMAX network simulation scheduling algorithm with homogeneous method algorithm and hybrid algorithm. Representatives on the homogeneous algorithms method will use Weighted Fair Queuing scheduling algorithm (WFQ) and Deficit Round Robin (DRR), whereas the hybrid method algorithm using merger between DRR and WFQ scheduling algorithm. Testing the performance of the scheduling algorithm is performed by comparing the QoS class into 5 types, namely WiMAX UGS, rtPS, nrtPS, ertPS and Best Effort. From the test results, a hybrid algorithm provides better QoS value when compared with homogeneous algorithm. Hybrid algorithm is very suitable when applied to network conditions that have traffic with varying packet data, because it can produce a high throughput, and can result in delay and jitter are also low.

Keywords - QoS, homogeneous algorithm, hybrid algorithm

I. PENDAHULUAN

Perkembangan teknologi komunikasi nirkabel (*wireless*) dewasa ini sangat pesat sekali. Teknologi nirkabel berkembang dari AMPS, GSM, CDMA hingga layanan 3G yang saat ini telah dapat melayani tidak hanya untuk layanan suara (*voice*) tetapi juga untuk data, gambar dan video. Dengan suksesnya penggunaan teknologi komunikasi nirkabel telah mendorong pengembangan teknologi menuju ke arah yang lebih baik dalam hal kapasitas, kecepatan,

kualitas, dan lebar pita (*bandwidth*). Salah satu teknologi nirkabel yang diperkirakan banyak digunakan untuk masa sekarang dan masa depan adalah WiMaX (*Worldwide Interoperability for Microwave Access*). IEEE 802.16e merupakan standar yang digunakan pada jaringan WiMaX, dan diharapkan muncul sebagai teknologi *Broadband Wireless Access* (BWA) yang dapat memberikan kecepatan tinggi kepada pelanggan.[1]

Salah satu keunggulan WiMaX dibandingkan teknologi internet nirkabel yang telah ada adalah kemampuan memberikan kepastian *Quality of service* (QoS). Terdapat 5 tipe kelas layanan QoS yang disediakan oleh WiMaX yaitu *Unsolicited Grand Service* (UGS), *Real time Polling Service* (rtPS), *Non Real Time Polling Service* (nrtPS), *Extended Real time Polling Service* (ErtPS), dan *Best Effort* (BE).[5]

Dengan mengacu pada beberapa sumber, terdapat beberapa algoritma penjadwalan, diantaranya adalah *Weight Fair Queuing* (WFQ) dan *Deficit Round Robin* (DRR). Algoritma-algoritma penjadwalan tersebut merupakan tipe *homogenous algorithm*. [2] Pada penelitian ini akan mencoba melakukan evaluasi kinerja algoritma penjadwalan dengan metode *hybrid algorithm*. Metode *hybrid algorithm* yang akan diimplementasikan pada penelitian ini adalah penggabungan antara algoritma penjadwalan DRR dan WFQ. Dengan adanya pembagian antrian data pada *Hybrid Algorithm*, yaitu paket data *realtime* ditangani oleh DRR dan paket data *non-realtime* ditangani oleh WFQ, maka akan dapat didapatkan hasil *throughput* yang tetap tinggi, *delay* yang rendah dan juga *jitter* yang rendah baik pada paket data *realtime* maupun paket data *non-realtime*. [6] Simulasi jaringan *mobile WiMaX* pada penelitian ini akan menggunakan *Network Simulator-2*

Penggunaan *homogenous algorithm* hanya optimal diterapkan pada kondisi-kondisi tertentu. [2] Melatarbelakangi hal tersebut maka perlu diterapkan *hybrid algorithm* untuk dapat saling melengkapi keunggulan dari setiap jenis algoritma penjadwalan pada kondisi tertentu.

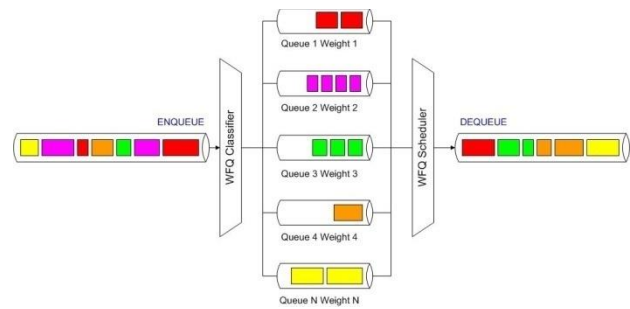
II. METODOLOGI

A. Algoritma Penjadwalan WIMAX

Algoritma penjadwalan pada BS *scheduler* tidak ditetapkan pada IEEE Std. 802.16. Hal ini memungkinkan para peneliti untuk mengembangkan algoritma penjadwalan yang optimal untuk diterapkan pada MAC layer WiMaX. Beberapa algoritma penjadwalan yang akan dibahas pada jurnal ini adalah *Weight Fair Queuing* (WFQ), *Deficit Round Robin* (DRR), dan *hybrid algorithm*.

- *Weight Fair Queuing* (WFQ)

WFQ digunakan untuk menentukan jumlah paket data yang diproses pada suatu waktu tertentu dan juga mengatur kapasitas kanal ketika pengalokasian *bandwidth* kepada SS. Penentuan jumlah bobot tergantung pada jenis layanan dan ukurannya. WFQ bekerja seperti memiliki beberapa gerbang pintu. Ketika paket data sampai, data tersebut akan diklasifikasi oleh *classifier* kemudian akan diteruskan ke salah satu pintu. Pada *classifier*, paket data dikelompokkan berdasarkan jenis layanan, ukuran, sumber, dan tujuan. Masing-masing pintu memiliki bobot yang berbeda-beda, artinya bobot tiap pintu disusun berdasarkan aplikasi yang berbeda-beda sesuai dengan QoS yang dibutuhkan [1].



Gambar 1. *Weight fair queuing*

Berikut ini adalah *pseudo code* dari algoritma penjadwalan WFQ :

Algorithm 2-1. Pseudo-code of WFQ

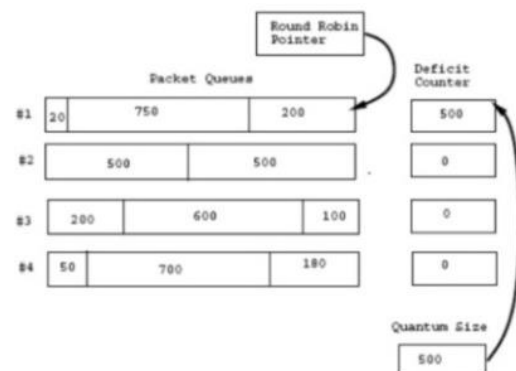
```

1  upon arrival of packet k
2    if (queue_length(queue_id(k)) < total_packet(wfq_
3      list, queue_id(k)))
4      drop k
5    end if
6    finish_time_k = computefinish_time(k,
7      queue_weight(k))
8    push(wfq_list, k, finish_time_k)
9
10 while (length(wfq_list) > 0)
11   p = pop(wfq_list)
12   send(p)
13 end while

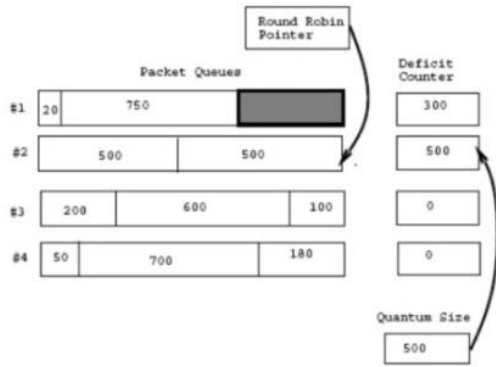
```

- *Deficit Round Robin* (DRR)

Pada algoritma DRR (*Deficit Round Robin*) urutan prioritas dilihat dari *deficit counter*-nya, apabila ukuran paket lebih kecil daripada *deficit counter* maka paket akan dikirim dan apabila ukuran paket lebih besar daripada *deficit counter* maka paket tidak akan dikirim [1].



Gambar 2. *Deficit round robin* (1)



Gambar 3. Deficit round robin (2)

Dari gambar 2 menunjukkan saat paket dari antrian #1 akan dilayani, nilai *deficit counter* antrian #1 diisi dengan *quantum* waktu. Karena ukuran paket pertama dari antrian #1 kurang dari *deficit counter*, maka paket tersebut dikirim. Gambar 3 menunjukkan kondisi ketika paket pertama dari antrian #1 telah dikirim. Pointer dari antrian akan dipindahkan ke antrian #2 dan *deficit counter* antrian #1 akan dikurangi dengan besarnya paket yang telah dikirim. Algoritma DRR efektif untuk diterapkan pada *datagram network* dengan jumlah paket yang sangat beragam.

Berikut ini adalah *pseudo code* dari algoritma penjadwalan DRR :

Algorithm 2-1. Pseudo-code of DRR

```

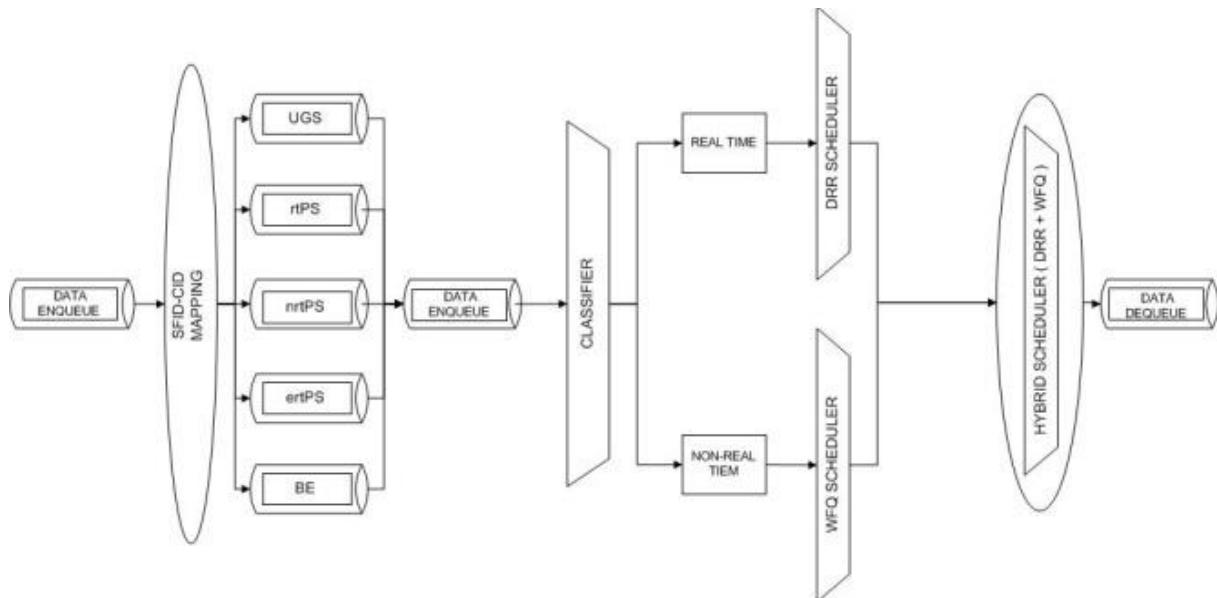
1  upon arrival of packet k
2    enqueue(k, queue[flow_id(k)])

3  while (length(queue[])>0)
4    q = next_round(queue)
5    while (size(head(q))>quantum(q))
6      p = pop(q)
7      quantum(q) = quantum(q) - size(p)
8      send(p)
9    end while
10   quantum(q) = quantum(q) + quantum_const send(p)
11  end while

```

- *Hybrid Algorithm*

Hybrid Algorithm pada penelitian ini merupakan gabungan antara algoritma WFQ dan algoritma DRR. Algoritma WFQ sangat cocok untuk mengeksekusi aplikasi *non real-time* (nrtPS, dan BE), sedangkan algoritma DRR memberikan hasil yang sangat baik untuk aplikasi-aplikasi *real-time* (UGS, ertPS dan rtPS).



Gambar 4. Hybrid algorithm

Dari gambar tersebut data antrian akan diproses ke dalam 2 proses algoritma penjadwalan sekaligus. Pada tahap awal data *enqueue* akan diklasifikasi berdasarkan aplikasinya, termasuk aplikasi *real-time* atau *non real-time*. Aplikasi-aplikasi *real-time* memiliki prioritas QoS terhadap *delay*, sedangkan aplikasi *non real-time* lebih menekankan aspek QoS terhadap *throughput* dan *packet loss*. Oleh sebab itu, aplikasi *real-time* akan dieksekusi terlebih dahulu oleh DRR *scheduler*, baru selanjutnya akan diteruskan kepada WFQ *scheduler*. Diharapkan dengan metode

hybrid algorithm tersebut dapat meningkatkan kinerja sistem dengan perbaikan *delay* pada aplikasi-aplikasi *real-time* dan perbaikan *throughput* dan *packet loss* pada aplikasi-aplikasi *non real-time* [2].

Berikut ini adalah *pseudo code* dari algoritma penjadwalan WFQ – DRR (*Hybrid*) :

Algorithm 2-1. Pseudo-code of DRR-WFQ

```

1  upon arrival of packet k
2    if (pkt_qos(k)==nrtps || pkt_qos(k)==be)

```

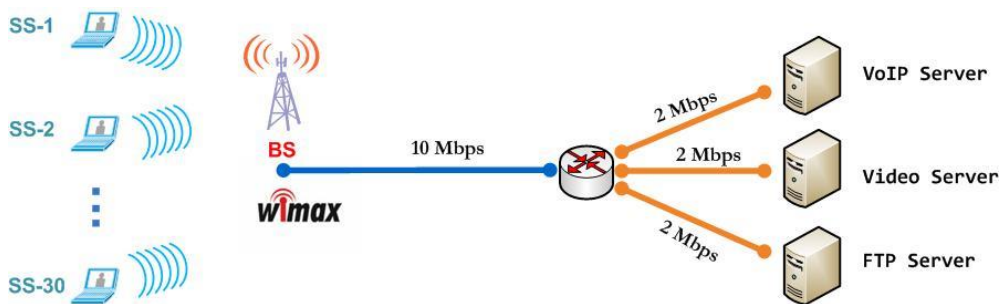
```

3     enqueue(k, queue[pkt_qos(k)][flow_id(k)])
4     else if (pkt_qos(k)==rtps || pkt_qos(k)==ertps ||
5         pkt_qos(k)==ugs)
6         finish_time_k = computefinishstime(k, queue_
7             weight(k))
8         push(wfq_list[pkt_qos(k)], k, finish_time_k)
9
10    for i in drr_list[ugs], drr_list[rtps], drr_list[ertps]
11        while(length(i[])>0)
12            q = next_round(i)
13            while(size(head(q))>quantum(q))
14                p = pop(q)
15                quantum(q) = quantum(q) - size(p)
16                send(p)
17            end while
18            quantum(q) = quantum(q) + quantum_const
19        end while
20    end for
21
22    for i in wfq_list[nrtps], wfq_list[be]
23        while(length(i)>0)
24            p = pop(i)
25            send(p)
26        end while
27    end for

```

B. Konfigurasi Jaringan

Jaringan Wimax terdiri dari dua komponen utama yaitu *Subscriber Station (SS)* dan *Base Station (BS)*. SS adalah perangkat Wimax yang ada pada sisi pelanggan. Ketika suatu permintaan koneksi disetujui, suatu layanan dengan parameter QoS tertentu akan dibuat. Layanan penjadwalan adalah mekanisme penanganan data untuk mendukung *MAC scheduler* untuk sebuah koneksi *data transport*. *Scheduler* akan mengkalkulasikan *throughput*. DL disiarkan dan *scheduler* mengisi setiap *burst* didasarkan pada parameter *frame QoS* pada antrian. Penjadwalan UL menggunakan skema yang lebih kompleks karena memerlukan koordinasi antara BS dan setiap individu SS. Pada penelitian ini jumlah total *Subscriber station (SS)* yang digunakan adalah 30 SS.



Gambar 5 Topologi simulasi

Parameter – parameter performansi yang digunakan dalam pengujian penelitian ini adalah :

- *Throughput*

Nilai *throughput* system ditentukan dengan :

$$Throughput = \frac{\text{Jumlah bit data diterima benar}}{\text{waktu pengiriman bit}} \text{Bps} \dots\dots\dots (1)$$

Throughput sistem adalah jumlah bit benar yang diterima dibandingkan dengan waktu pengiriman bit.

- *Packet loss*

Packet loss didefenisikan sebagai kegagalan transmisi paket mencapai tujuannya. Kegagalan paket tersebut dapat disebabkan terjadinya *overload traffic* di dalam jaringan akibat keterbatasan memori yang dimiliki *network node*. Selain itu kegagalan juga disebabkan karena tabrakan (*congestion*) dalam jaringan yang disebabkan oleh antrian yang telah melewati batas jumlah paket yang yang dapat ditampung dalam *buffer*.

Nilai *packet loss* dapat dinyatakan sebagai :

$$Packet\ loss = \frac{\text{jml bit dikirim}-\text{jml bit diterima}}{\text{jml bit dikirim}} \times 100\% \dots\dots\dots (2)$$

Di dalam implementasi jaringan IP nilai *packet loss* diharapkan mempunyai nilai yang minimum.

- *Delay*

Delay merupakan rentang waktu antara waktu kedatangan paket di antrian hingga keluarnya paket dari antrian dan siap ditransmisikan :

$$delay(n) = T_{out}(n) = \text{servicing time} + T_{in}(n) \dots (3)$$

dimana :

$T_{out}(n)$ = waktu data ke-n keluar antrian dan siap ditransmisi

$T_{in}(n)$ = waktu data ke-n masuk antrian

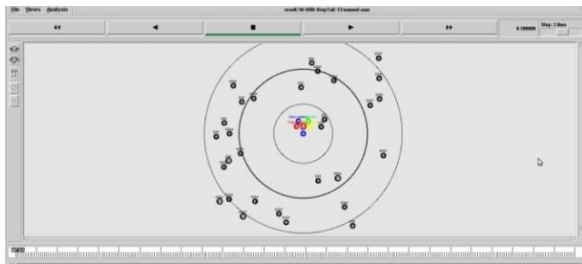
- *Jitter*

Jitter merupakan variasi *delay* antar paket yang terjadi pada jaringan IP. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP.

III. PEMBAHASAN

Skenario pengujian dilakukan dengan pengambilan data pada *user mobile* menggunakan metode *random seed*. *Random seed* ini bertujuan untuk *generate random number* saat simulasi, yang digunakan untuk penentuan lokasi *mobile node* (setelah dibulatkan). Setiap *random seed* akan *generate* deretan

random number yang sama, meskipun digunakan dalam proses simulasi yang berbeda.



Gambar 6. Skenarion pengujian random seed 13

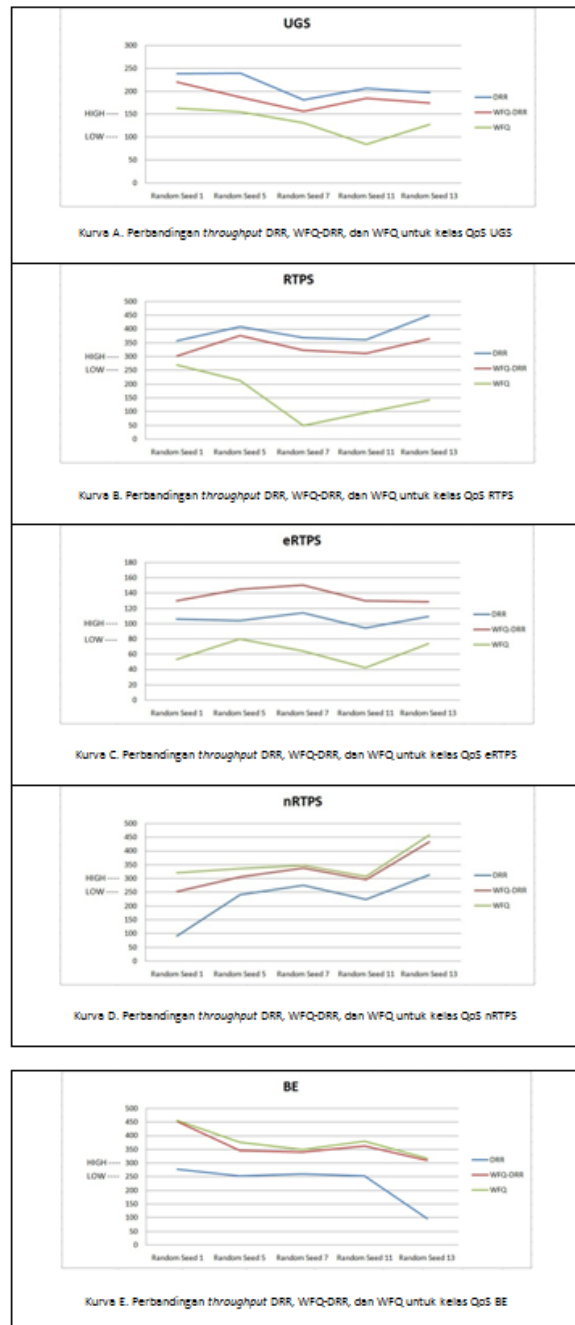
Sebagaimana yang telah dijelaskan diatas, ada 5 jenis trafik dari IEEE 806.16e yaitu UGS, RTPS, eRTPS, nRTPS, dan BE. Pengujian dilakukan dengan membagi jumlah mobile node dengan perbandingan mobile node untuk kelas QoS UGS; RTPS; eRTPS; nRTPS; BE adalah 6; 6; 6; 6; 6 user SS. Jumlah total Subscriber station (SS) yang digunakan adalah 30 SS. Parameter yang akan diuji adalah delay, throughput, packet loss, dan Jitter.

A. Throughput

Pada Gambar 7 ditunjukkan nilai throughput untuk 5 jenis kelas QoS, yaitu UGS, RTPS, nRTPS, eRTPS, dan BE. Nilai throughput pada setiap kelas QoS tersebut hasilnya akan dilakukan perbandingan antara homogeneous algorithm (WFQ dan DRR) dengan Hybrid algorithm (WFQ+DRR). Pengambilan data pada setiap algoritma penjadwalan dilakukan pada kondisi mobile. Dimana skenario trafik jaringan untuk kondisi mobile dilakukan secara random seed yang terbagi ke dalam 5 jenis, yaitu random seed 1, random seed 5, random seed 7, random seed 11, dan random seed 13.

Pada Gambar 7 bagian kurva A dan kurva B ditunjukkan bahwa pada kelas QoS UGS dan RTPS, algoritma penjadwalan DRR mendapatkan nilai throughput yang tinggi, yaitu mencapai 250 Kbps untuk kelas QoS UGS dan mencapai 500 Kbps untuk kelas QoS RTPS. Hasil yang baik ini juga ditandai dengan tingginya kurva tersebut. Namun kelas-kelas QoS yang bersifat realtime tersebut berjalan kurang baik jika menggunakan algoritma penjadwalan WFQ. Pada algoritma penjadwalan WFQ terjadi penurunan throughput pada paket data realtime tersebut, yaitu ditandai dengan rendahnya nilai kurva seperti yang terlihat pada Gambar 7 bagian kurva A dan kurva B. Disisi lain, untuk paket data non-realtime (nRTPS dan BE) algoritma penjadwalan WFQ dapat menghasilkan nilai throughput yang tinggi, yaitu mencapai 500 Kbps. Hal ini juga ditunjukkan dengan adanya kenaikan nilai throughput pada Gambar 7 bagian kurva D dan kurva E. Sedangkan untuk algoritma hybrid selalu menunjukkan nilai throughput yang relatif baik dan stabil pada kelima kelas QoS tersebut. Seperti yang terlihat pada Gambar 7, algoritma hybrid menghasilkan rata-rata nilai yang tinggi pada semua paket data, baik yang bersifat realtime maupun non-realtime. Dengan adanya penilaian ini, dapat disimpulkan bahwa algoritma hybrid dapat

meghasilkan throughput yang baik jika dijalankan pada sistem yang memiliki trafik data yang beragam, yaitu paket data realtime maupun non-realtime.



Gambar 7. Perbandingan throughput

B. Delay

Delay merupakan rentang waktu antara waktu kedatangan paket di antrian hingga keluarnya paket dari antrian dan siap ditransmisikan.

Minimum delay request untuk kelas QoS UGS adalah < 250 ms, kelas QoS RTPS < 600 ms, kelas QoS eRTPS < 80 ms, kelas QoS nRTPS < 600 ms, dan kelas QoS BE < 600 ms. Apabila hasil delay yang didapat memenuhi nilai minimum tersebut maka mengindikasikan bahwa sistem berjalan dengan baik. Kelas QoS UGS, RTPS, dan eRTPS merupakan paket

data bersifat *realtime* yang rentan terhadap delay. Sehingga diperlukan algoritma penjadwalan yang dapat memberikan *delay minimum* untuk paket-paket data *realtime* tersebut.



Gambar 8. Perbandingan *delay*

Algoritma DRR dan algoritma *hybrid* dapat memenuhi permintaan tersebut, yaitu dapat melayani paket-paket data *realtime* dengan menghasilkan nilai *delay* yang rendah. Seperti yang terlihat pada gambar 8 bagian kurva A, kurva B, dan kurva C, algoritma DRR dan algoritma *hybrid* menghasilkan rata-rata nilai yang rendah yaitu nilai *delay* < 120 ms untuk kelas QoS UGS, nilai < 240 ms untuk kelas QoS RTPS, dan nilai < 30 ms untuk kelas QoS eRTPS. Sedangkan hasil *delay* yang baik untuk paket-paket data *non-realtime* dihasilkan oleh algoritma WFQ dan

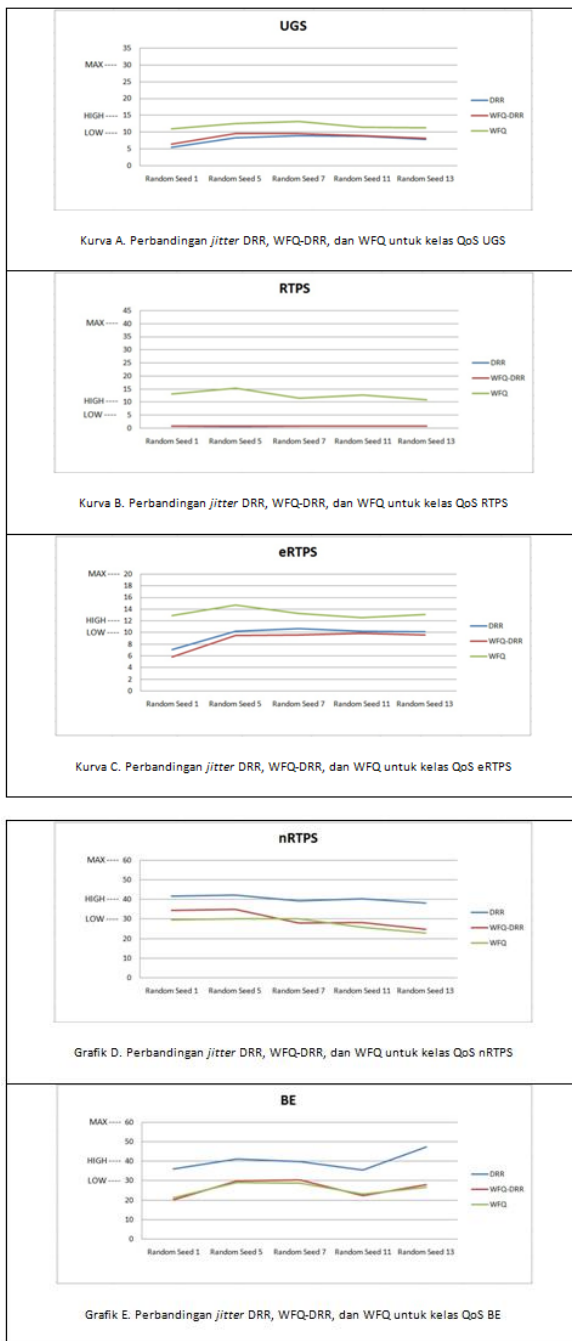
algoritma *hybrid*. Nilai yang banyak muncul pada kurva perbandingan untuk kedua algoritma tersebut adalah relatif rendah, yaitu menghasilkan nilai *delay* < 450 ms baik untuk kelas QoS nRTPS maupun kelas QoS BE. Hal ini menunjukkan bahwa untuk paket data *non-relatime*, algoritma WFQ dan *hybrid* dapat menghasilkan nilai *delay* yang sedikit lebih baik daripada algoritma DRR. Sehingga dapat ditarik kesimpulan, algoritma *hybrid* dapat memberikan hasil *delay* yang *minimum* untuk paket-paket data *realtime* tanpa mengesampingkan hasil *delay* pada paket-paket data *non-realtime*

C. Jitter

Jitter merupakan variasi *delay* antar paket, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan.. Besar nilai *jitter* dipengaruhi oleh variasi beban trafik dan besar tumbukan antar paket (*congestion*) yang ada di dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai QoS akan semakin turun. Sistem jaringan dikatakan dapat berjalan dengan baik apabila nilai *jitter* yang dihasilkan di dalam sistem tersebut sangat rendah. Nilai *jitter* yang dihasilkan pada penelitian ini masih memenuhi standar minimum yang diminta, yaitu kurang dari 100 ms. Hasil *jitter* pada setiap algoritma penjadwalan memberikan hasil yang relatif baik dan tidak terjadi perbedaan nilai *jitter* yang *significant*. Seperti halnya hasil pada *delay*, untuk paket – paket data *real-time* algoritma penjadwalan DRR masih lebih unggul untuk menghasilkan nilai *jitter* yang paling minimum. Namun, hasil ini tidak berbeda jauh dengan nilai *jitter* yang dihasilkan algoritma gabungan antara DRR dan WFQ. Hal ini disebabkan karena algoritma tersebut dapat mengatur dengan baik selang waktu kedatangan antar paket dengan melakukan pembagian merata untuk pelayanan paket-paket data *realtime* maupun paket-paket data *non-realtime*. Oleh karena itu pada paket-paket data *non-realtime*, algoritma gabungan ini juga menghasilkan nilai yang baik. Pada Gambar 9 ditunjukkan gambar perbandingan algoritma penjadwalan DRR, WFQ-DRR, dan WFQ untuk kelas QoS UGS, RTPS, nRTPS, eRTPS, dan BE.

Nilai *jitter* yang dihasilkan oleh algoritma DRR dan algoritma *hybrid* sangatlah baik, terutama untuk paket-paket data *realtime* (UGS, RTPS, eRTPS). Hal ini ditunjukkan juga pada Gambar 9 bagian kurva A dan kurva B, yaitu pada kedua algoritma tersebut didapatkan nilai yang rendah untuk kelas QoS UGS dan eRTPS. Sedangkan Gambar 9 bagian kurva C, menunjukkan bahwa kelas QoS RTPS didominasi oleh nilai-nilai yang sangat rendah. Pada kedua algoritma tersebut, nilai *jitter* untuk paket data UGS dan eRTPS adalah kurang dari 12 ms, sedangkan nilai *jitter* untuk kelas QoS nRTPS adalah kurang dari 10 ms. Sedangkan pada algoritma WFQ menghasilkan hasil

jitter yang kurang baik jika dijalankan pada paket data *realtime*.



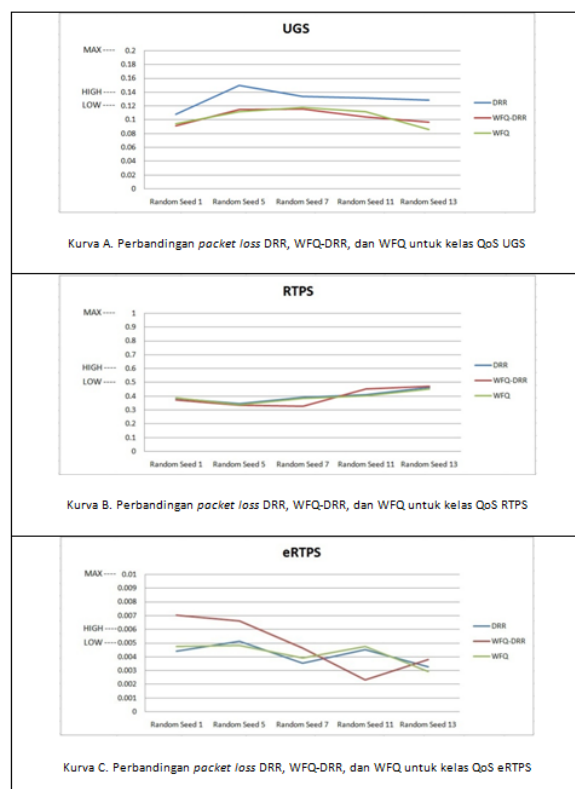
Gambar 9. Perbandingan *jitter*

Nilai *jitter* yang muncul pada algoritma tersebut relatif tinggi untuk kelas QoS UGS dan RTPS, sedangkan untuk kelas QoS eRTPS didominasi oleh nilai kurva yang tinggi. Akan tetapi untuk paket-paket data *non realtime*, algoritma WFQ dan algoritma *hybrid* menghasilkan nilai *jitter* yang sedikit lebih baik jika dibandingkan dengan algoritma DRR. Walaupun perbedaan nilai *jitter* paket data *realtime* dan *non-realtime* yang dihasilkan antara algoritma WFQ dan DRR tidak terlalu *significant*, namun hal tersebut telah menunjukkan bahwa adanya perbedaan keadilan dalam pelayanan suatu paket. Oleh karena itu,

algoritma *hybrid* merupakan solusi dari permasalahan tersebut, yaitu dapat memberikan hasil *jitter* yang relatif rendah dan dapat melayani dengan baik setiap paket data yang datang, baik itu paket data *realtime* maupun paket data *non-realtime*.

D. Packet Loss

Packet Loss menunjukkan banyaknya jumlah paket yang hilang. *Packet Loss* terjadi ketika satu atau lebih paket data yang melewati suatu jaringan gagal mencapai tujuannya. Paket yang hilang atau drop dapat menurunkan performansi terutama pada aplikasi seperti teknologi *streaming*, *Voice over IP (VoIP)*, *online gaming*, dan *video conference*. Akan tetapi, sangat penting untuk diketahui bahwa paket hilang tidak selalu mengindikasikan adanya suatu permasalahan dalam jaringan. Jika besarnya paket hilang masih dapat diterima oleh tujuan, maka terjadinya paket hilang bukan suatu permasalahan. Pada penelitian ini dilakukan pembahasan analisa *packet loss* hanya pada paket-paket data *realtime*. Hal ini disebabkan karena paket-paket data *realtime* rentan terhadap adanya paket yang hilang. Sistem jaringan berjalan dengan baik apabila tingkat keberhasilan pengiriman paket mencapai nilai lebih dari 99 % atau dengan kata lain data yang hilang kurang dari 1 %. Hasil dari pengambilan data *packet loss* menunjukkan nilai yang masih memenuhi QoS *requirements*, yaitu pengiriman paket mencapai nilai lebih dari 99 %.



Gambar 10. Perbandingan *packet loss*

Ketiga algoritma yang disimulasikan (DRR, WFQ, dan *hybrid*) dapat memenuhi standar minimum *packet loss requirements* yang diminta, yaitu < 0.2 % untuk

Kelas QoS UGS, < 1 % untuk kelas QoS RTPS, dan < 0.01 % untuk kelas QoS eRTPS. Nilai *packet loss* untuk ketiga paket data (UGS, RTPS, dan eRTPS) masih lebih unggul jika menggunakan algoritma WFQ daripada menggunakan algoritma DRR maupun algoritma *hybrid*. Sesuai dengan gambar 10 terlihat bahwa algoritma WFQ memiliki rata-rata nilai yang rendah, sedangkan algoritma DRR dan *hybrid* memiliki rata-rata nilai yang tinggi. Akan tetapi, pada gambar 10 terlihat juga bahwa algoritma *hybrid* memberikan hasil yang kurang maksimal untuk kelas QoS eRTPS. Hal ini disebabkan karena kelas QoS eRTPS memiliki trafik dengan ukuran paket yang beragam, sehingga data yang hilang dapat terjadi saat pembagian layanan antara paket data *realtime* dengan *non-realtime*. Oleh karena itu penggunaan algoritma *hybrid* memiliki potensi yang lebih besar terhadap hilangnya paket data.

IV. PENUTUP

A. Kesimpulan

Berikut ini adalah beberapa kesimpulan yang dapat diambil terkait dengan pengerjaan penelitian ini:

1. Algoritma *Weighted Fair Queueing* (WFQ) cocok diterapkan untuk kondisi jaringan dimana *error state* sering terjadi, sehingga jaminan akan keutuhan paket yang diterima sangat tinggi. Hal ini karena algoritma WFQ mempertimbangkan sisi kualitas link dalam menentukan penjadwalan pengiriman paket. Algoritma *Weighted Fair Queueing* (WFQ) cocok untuk paket-paket data *non-realtime* seperti nRTPS dan BE.
2. Algoritma *Deficit Round Robin* (DRR) cocok diterapkan untuk kondisi jaringan yang memiliki trafik dengan ukuran paket yang beragam dan memiliki trafik data dari jenis kelas layanan yang berprioritas tinggi, yaitu UGS, rTPS, dan eTPS. Ukuran paket data yang selalu sama dan tidak rentan terhadap *delay* membuat parameter *deficit counter* pada DRR tidak bermanfaat.
3. Algoritma *Hybrid* yaitu yang dibahas pada penelitian ini merupakan gabungan antara algoritma WFQ dan algoritma DRR. Algoritma *Hybrid* (WFQ + DRR) cocok diterapkan untuk kondisi jaringan yang memiliki trafik dengan paket data yang bervariasi, yaitu adanya paket data *realtime* dan paket data *non-realtime* yang datang secara simultan. Hal ini pada algoritma *hybrid* dilakukan pembagian antrian data, yaitu paket data *realtime* ditangani oleh DRR dan paket data *non-realtime* ditangani oleh WFQ.
4. Dari pengujian yang dilakukan memakai kerangka uji terhadap algoritma-algoritma tersebut, dapat diambil kesimpulan sebagai berikut :
 - Paket data *realtime*
Untuk nilai *throughput*, nilai *delay*, dan nilai *jitter*, hasil yang baik diperoleh dari algoritma DRR dan algoritma *hybrid*. Dengan kedua

algoritma tersebut didapat *throughput* yang tinggi, *delay* yang rendah yaitu mencapai < 60 ms, dan *jitter* yang rendah yaitu mencapai < 10 ms. Sedangkan pada algoritma WFQ didapatkan hasil yang kurang bagus.

- Paket data *non-realtime*
Untuk nilai *throughput*, nilai *delay*, dan nilai *jitter*, hasil yang baik diperoleh dari algoritma WFQ dan algoritma *hybrid*. Dengan kedua algoritma tersebut didapat *throughput* yang tinggi, *delay* yang rendah yaitu mencapai < 300 ms, dan *jitter* yang rendah yaitu mencapai < 40 ms. Sedangkan pada algoritma DRR didapatkan hasil yang kurang bagus.
 - *Packet Loss*
Hasil data *packet loss* dari ketiga algoritma penjadwalan tersebut menunjukkan hasil yang relatif bagus, yaitu tingkat keberhasilan pengiriman paket data lebih dari 99 %, atau dengan kata lain data yang hilang < 1 %. Namun apabila dilakukan perbedaan tingkatan maka urutan terbaik diperoleh dari algoritma WFQ, kemudian disusul oleh algoritma DRR, dan yang terakhir adalah algoritma *hybrid*.
5. Permasalahan yang sering kali muncul ketika menggunakan *homogeneous algorithm* adalah pembagian pada saat melayani paket data beragam (*realtime* maupun *non-realtime*) yang datang secara simultan. Adanya ketimpangan pelayanan terhadap suatu paket tertentu mengakibatkan terjadinya penurunan nilai QoS.
 6. Permasalahan dapat muncul ketika menggunakan *hybrid algorithm* adalah rentan terjadinya data yang hilang pada saat pembagian layanan antara paket data *realtime* dengan *non-realtime*. Semakin tinggi trafik didalam antrian akan dapat mengakibatkan nilai *packet loss* yang semakin tinggi.

B. Saran

1. Untuk analisis lebih lanjut, dapat dilakukan pengujian terhadap gabungan algoritma penjadwalan (*hybrid algorithm*) dari jenis algoritma yang lainnya, seperti, EDF, PI, WRR, dan lain-lain. Gabungan algoritma penjadwalan juga dapat dilakukan setelah melakukan modifikasi dari satu jenis algoritma, misalnya MWFQ dengan MEDF. Gabungan beberapa jenis algoritma memungkinkan menghasilkan nilai metrik yang lebih baik.
2. Terdapat dua metrik yang dinilai dapat mengukur kualitas suatu algoritma penjadwalan namun tidak dilakukan pengukuran pada penelitian kali ini. Kedua metrik tersebut adalah *frame utilization* dan *fairness index*. Implementasi kedua metrik tersebut dapat dilakukan dengan mengubah struktur *trace file* hasil simulasi. Dengan mengimplementasikan kedua metrik tersebut, penilaian terhadap kualitas algoritma penjadwalan dapat dilakukan dengan lebih lengkap.

DAFTAR PUSTAKA

- [1] A. Jeffrey G, Fundamentals of WiMAX Understanding Broadband Wireless Networking, Prentice Hall, 2007..
- [2] Ali, N.A., Dhrona, P., and Hassanein, H., A performance study of uplink scheduling algorithms in point-to-multipoint WiMAX networks, Computer Communications 32 (2009) 511–521, Elsevier B.V., 18 September 2008.
- [4] C. Kwang-Cheng, J. Roberto, and B. de Marca, Mobile WiMAX, IEEE Press, England, 2008.
- [5] Chowdhury, P., and Misra, I.S., A Fair and Efficient Packet Scheduling Scheme for IEEE 802.16 Broadband Wireless Access Systems, International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.1, No.3, September 2010.
- [6] D. Pratik, A Performance Study of Uplink Scheduling Algorithms in Point to Multipoint WiMAX Networks, Thesis paper, Canada, December 2007.
- [7] Forouzan , Behrouz A., Data Communications and Networking, Fourth Edition, McGraw-Hill, New York, 2007.
- [8] G. Debalina, G. Ashima, M. Prasant, Scheduling in Multihop WiMAX Networks, ACM Sigmoblie Mobile Computing and Communication Vol 12 Issue 2 New York, April 2008.
- [9] J. Fanchun, A. Amrinder, and H. Jinho, Routing and Packet Scheduling for Throughput Maximization in IEEE 802.16 Mesh Networks, Washington.
- [10] Katz, D.M., Fitzek, F.H.P., WiMAX Evolution, John Wiley & Sons Ltd, United Kingdom. 2009.
- [11] Lipari, Giuseppe., Earliest Deadline First, Scuola Superiore Sant’Anna, Pisa -Italy. 2005.
- [12] Nagaraju, C., and Sarkar, M., A Packet Scheduling To Enhance Quality of Service in IEEE 802.16, Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I, San Franscisco, USA, October 20-22, 2009
- [13] Paschos. G.S., I. Papapanagiotou, C.G. Argyropoulos and S.A. Kotsopoulos, A Heuristic Strategy for IEEE 802.16 WiMAX scheduler for Quality of Service, 45th Congress FITCE 2006, Athens, Greece, 2006.

