

BAB IV

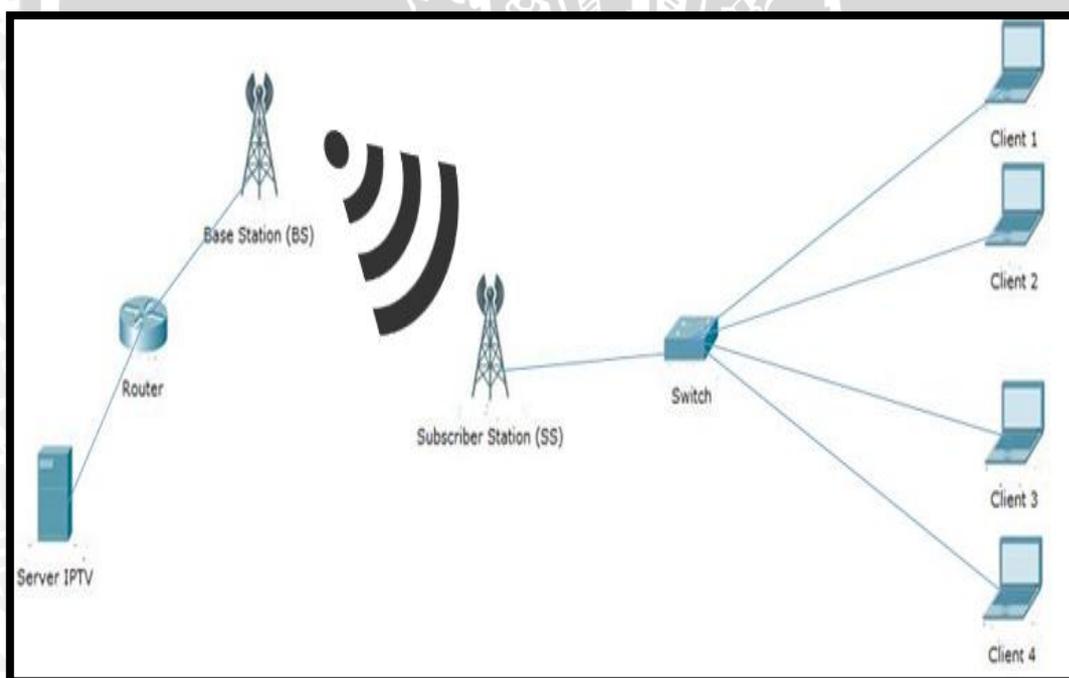
PEMBAHASAN DAN HASIL

4.1 Umum

Penelitian yang dilakukan adalah dengan rancang bangun jaringan IPTV melalui WiMAX dengan memvariasikan kelas layanan yang di sediakan oleh WiMAX IEEE 80216d. Adapun tahapan-tahapan adalah perancangan hingga pengujian rancang bangun sistem, tahapan melakukan analisis perhitungan data sekunder maupun pengambilan data primer meliputi delay, packet loss, dan throughput, kemudian membandingkan hasil pengujian dengan perhitungan teoritis.

4.2 Perancangan Blok Diagram

Perancangan menjelaskan tentang perangkat-perangkat yang digunakan dalam rancangan bangun jaringan seperti server, WiMAX, dan user itu sendiri.



Gambar 4.1 Konfigurasi Perancangan Jaringan

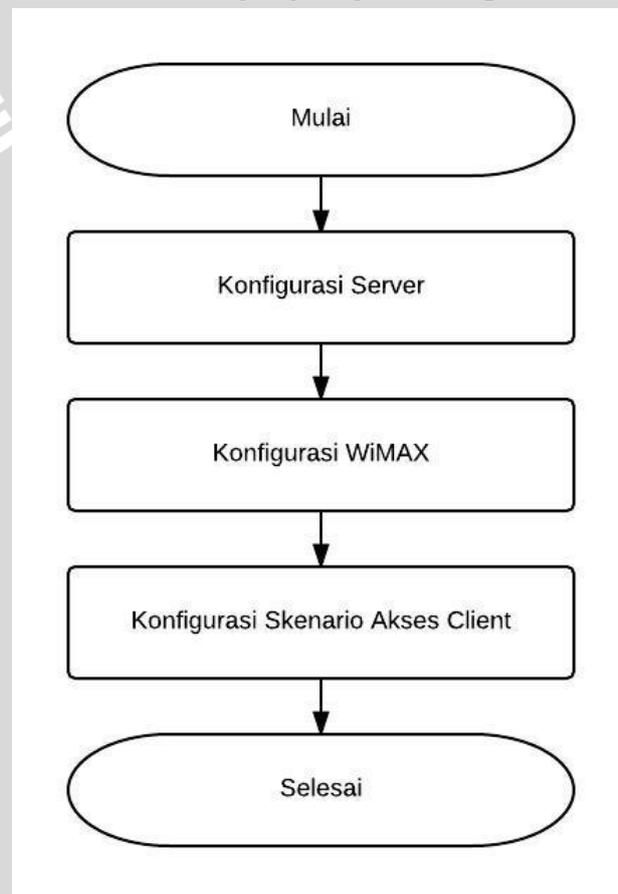
(Sumber: Perancangan)

Penelitian ini menggunakan sebuah *server* IPTV yang berada di Laboratorium Komputasi dan Jaringan yang dihubungkan langsung dengan perangkat WiMAX *Base Station* yang

berada di Laboratorium Telekomunikasi dengan kabel RJ-45. Kemudian *Base Station* dan *Subscriber Station* yang berjarak 10 meter berkomunikasi dengan gelombang radio, dan melalui sebuah perangkat *switch 4 user* yang masing-masing berupa perangkat *laptop* akan mengakses jaringan IPTV yang ditangkap oleh *Subscriber Station*.

4.3 Konfigurasi Jaringan

Konfigurasi jaringan yang dimaksud adalah konfigurasi jaringan secara umum untuk membangun konstruksi jaringan yang akan dibuat, pada Gambar 4.2 merupakan langkah-langkah yang harus dilakukan untuk membangun jaringan IPTV pada WiMAX.



Gambar 4.2 Mekanisme Konfigurasi Jaringan

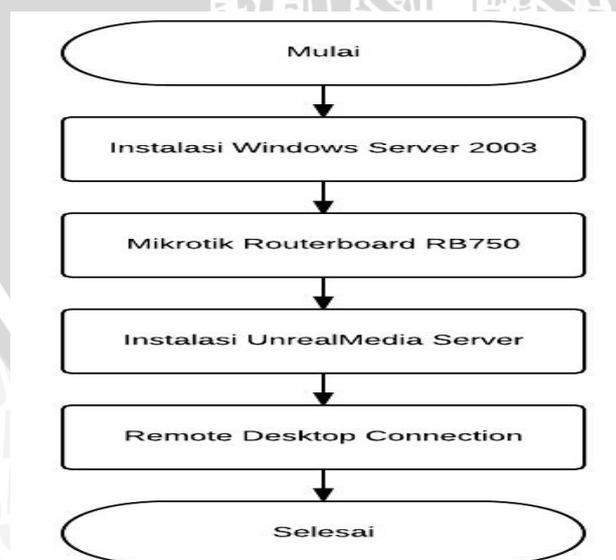
(Sumber: Perancangan)

Untuk membangun jaringan IPTV pada WiMAX hal yang harus dilakukan adalah mengkonfigurasi *server* yaitu mengisi konten serta mengatur IP agar dapat diakses oleh user, setelah itu melakukan konfigurasi WiMAX pada sisi *Base Station (BS)* dan *Subscriber Station (SS)*, kemudian mengatur user dengan mengakses IP yang telah ditentukan agar jaringan IPTV yang telah dibuat dapat diakses dengan baik.

4.3.1 Konfigurasi Server

Untuk melakukan konfigurasi *server*, berikut merupakan aspek-aspek yang harus diperhatikan dalam pembangunan *server* yang berada di Laboratorium Komputasi dan Jaringan sebagai mana dijelaskan pada gambar 4.3, yaitu:

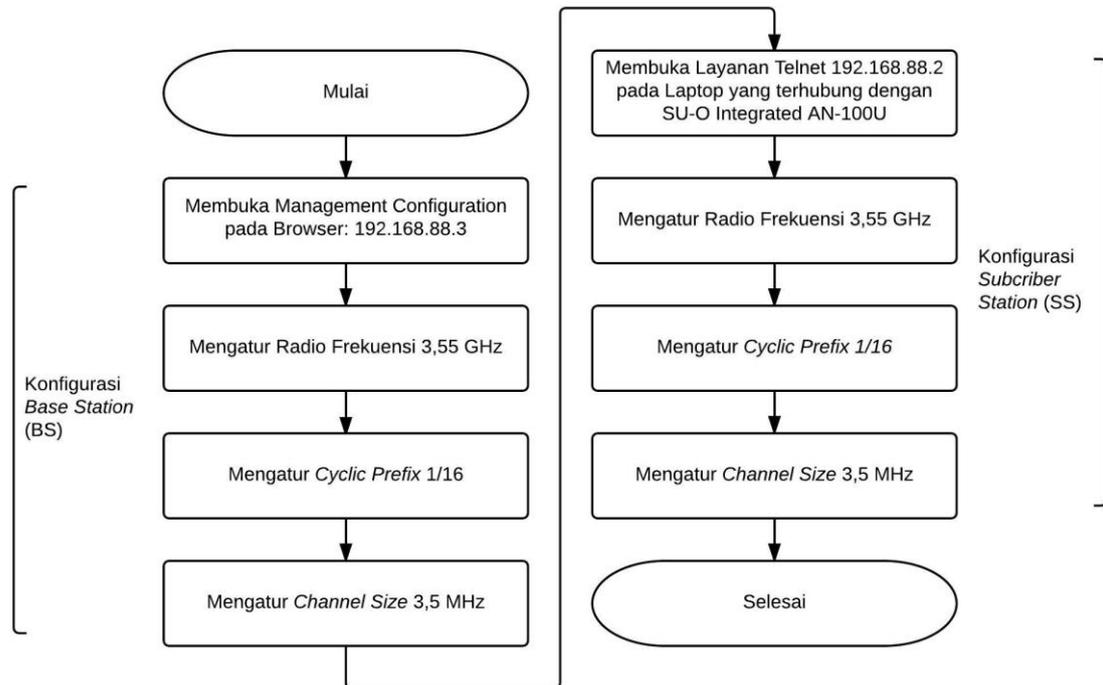
- a. Mekanik dan Elektrik
Menghubungkan masing-masing perangkat keras dengan media fisik, rak server, UPS.
- b. Instalasi Windows Server
Instalasi perangkat lunak Windows Server 2003 pada PC server sebagai media server yang mendukung layanan IPTV.
- c. Mikrotik Routerboard RB750
Mengkonfigurasi *interface Ethernet* 1 dari ISP dan Ethernet 2 menuju Client, DHCP Server, DHCP Client, IP DNS, IP Firewall, dan IP Client.
- d. Unreal Media Server
Perangkat lunak ini dipilih karena mendukung fitur IPTV, pada perangkat lunak ini dapat mengatur jenis protokol yang akan di lewatkan serta mengkonfigurasi IP.
- e. Remote Desktop Connection
Mengkonfigurasi komputer server sehingga dapat dikendalikan dengan perangkat komputer lain yang terhubung ke jaringan.



Gambar 4.3 Konfigurasi Server

(Sumber: Perancangan)

4.3.2 Konfigurasi WiMAX



Gambar 4.4 Mekanisme Konfigurasi WiMAX

Untuk mengkonfigurasi WiMAX perangkat yang harus diperhatikan adalah *Base Station* (BS) dan *Subscriber Station* (SS), gambar 4.4 menjelaskan langkah-langkah yang harus dilakukan untuk mengkonfigurasi BS dan SS. Pada perangkat BS operator dapat mengakses IP dari BS itu sendiri pada browser, kemudian mengatur radio frekuensi sebesar 3,55 GHz, *cyclic prefix* 1/16, dan *channel size* dengan nilai 3,5 MHz. Kemudian untuk mengkonfigurasi SS operator dapat membuka layanan *telnet* dengan IP SS itu sendiri dan mengatur radio frekuensi, *cyclic prefix*, dan *channel size* dengan nilai yang sama seperti yang telah diberikan pada saat konfigurasi BS.

4.3.2.1 Konfigurasi Perangkat *Base Station* (BS) RedMAX AN-100U

Sistem *Base Station* (BS) merupakan produk keluaran dari RedMAX yang memfasilitasi operator untuk dengan cepat merubah pengaturan yang disediakan oleh WiMAX dan juga berguna sebagai media penghubung dengan *Subscriber Station* (BS). Fitur

dari AN – 100 U terdapat pada gambar 4.4. fitur tersebut merupakan tampilan dari depan yang terdiri dari beberapa bagian antara lain *wireless port*, *time synchronization ports*, dan *wireless LED*.



Gambar 4.5 Perangkat *Base Station*

(Sumber: Perancangan)

Untuk membuka akses BS dengan cara menuliskan alamat IP yang diatur pada BS di browser. Setelah memasuki laman konfigurasi BS sebagaimana tertera pada gambar 4.

Langkah selanjutnya yang dilakukan antara lain:

- Konfigurasi Kelas Layanan (*Service Class*)

Pada konfigurasi *service class* digunakan untuk mengatur kelas layanan yang digunakan dan menentukan kapasitas kanal. Pada penelitian ini menggunakan kapasitas kanal sebesar 2 Mbps dengan mengubah-ubah jenis kelas layanan yang digunakan yaitu *real Time Polling Service* (rTPS), non real Time Polling Service (nrTPS), bE (best Effort), dan *unsolicited Grant Service* (uGS) dengan cara mengubah pada kolom *Sceduling Type*. Gambar 4.7 Menunjukkan tampilan konfigurasi service class.

Service Class Configuration

Add/Modify a Service Class

Service Class Name: uGS Traffic Priority: 7

Max Sustained Rate [bps]: 2000000 Min Reserved Rate [bps]: 2000000

Max Latency [ms]: 30 Fixed vs. Variable Sdu Ind: variableLength

Sdu Size [byte]: 0 Scheduling Type: unsolicitedGrantService

Req Tx Policy: noBroadcastBwReq(0) noPiggybackReq(2) noFragmentData(3)

noPHS(4) noSduPacking(5) noCrc(6)

Delete a Service Class (must not be used by SFs)

Service Class Name: bE Delete

Service Classes

Select: uGS ShowAll HideAll

SC Name	Traffic Prio.	MaxSTR	MinRR	MaxLat	Fixed vs Var. Sdu	Sdu Size	Sched. Type	ReqTxPol
bE	0	2000000	0	0	variableLength	0	bestEffort	4
rTPS	7	2000000	0	30	variableLength	0	realTimePollingService	4
nrTPS	0	2000000	0	0	variableLength	0	nonRealTimePollingService	4
uGS	7	2000000	2000000	30	variableLength	0	unsolicitedGrantService	4

Gambar 4.6 Konfigurasi Service Class

(Sumber: Perancangan)

- Konfigurasi Service Flow

Setelah terkonfigurasi di *service class*, BS akan terhubung dengan cara mengatur sistem *upstream* dan *downstream* di *service flow*. Seperti Gambar 4.8 ketika sudah aktif terhubung tampilan dapat terlihat seperti gambar dengan SS name “subscriber2”, *direction* “Downstream dan Upstream”. Jika sudah terhubung maka pada kolom SF State akan tertera “active” yang sebelumnya tertera “authorized”.

Service Flows Configuration

Next SFID	SS Name	Direction	SC Name	C/S Specification	
10	Subscriber	downstream	bE	802.3 Ethernet	Add

Delete SF (all associated Classifiers will be deleted)

Service Flow Identifier: 8 Delete

Service Flows

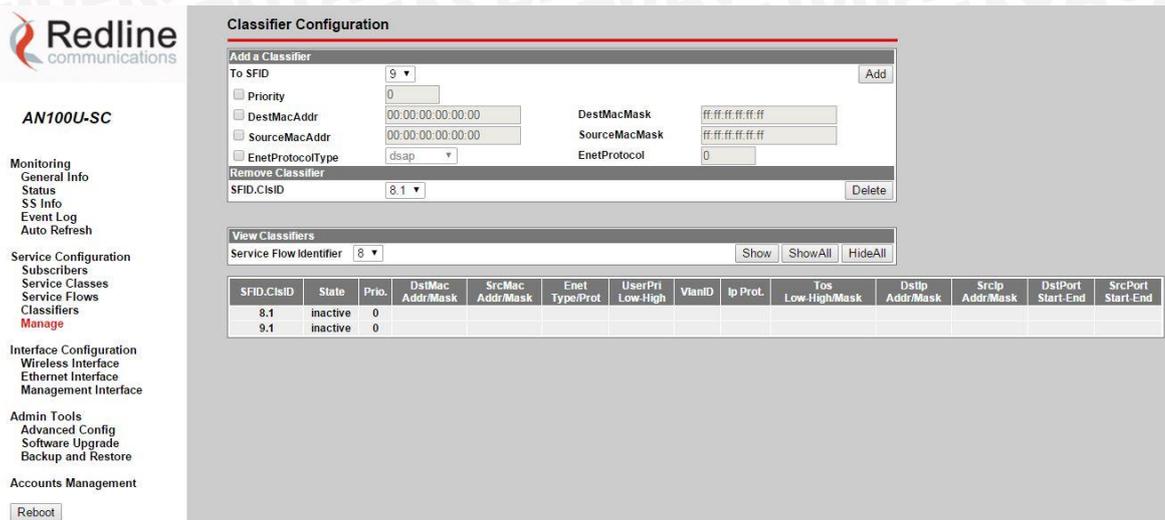
Select: 8 Template Edit ShowAll HideAll Enable Disable

SFID	SS Mac	SS Name	Direction	SC Name	SF State	Prov Time	C/S Specification	En/Dis
8	00:09:02:04:07:91	Subscriber 2	downstream	uGS	authorized	00:08:56	802.3 Ethernet	enabled
9	00:09:02:04:07:91	Subscriber 2	upstream	uGS	authorized	00:10:20	802.3 Ethernet	enabled

Gambar 4.7 Konfigurasi Service Flow
(Sumber: Perancangan)

- Konfigurasi Classifiers

Pada classifiers jaringan di atur dengan menambahkan nomer SFID yang sudah diatur sebelumnya di menu service flow. Gambar menunjukkan status inactive pada kolom state ketika BS telah tersambung dengan SS maka state akan berubah menjadi active.



Gambar 4.8 Konfigurasi Classifier

4.3.2.2 Konfigurasi Subscriber Station (SS) SU-O Integrated AN-100U

Subscriber station merupakan berfungsi sebagai penerima data dari base station. Pada WiMAX IEEE 802.16d atau Fixed WiMAX antena SS terdapat pada masing-masing rumah pelanggan. Terdapat lima SS yang berada di Laboratorium Telekomunikasi. Sedangkan untuk penelitian ini hanya menggunakan satu SS dengan panjang 7 meter dari BS. Gambar 4.5. merupakan tampilan dari port SS. Pada SS terdapat dua port, port pertama digunakan untuk menghubungkan SS dengan Antena sedangkan port kedua untuk menghubungkan SS dengan switch pelanggan.



Gambar 4.9 Perangkat Subscriber Station (SS)

(Sumber: Perancangan)

Konfigurasi SS dimulai dengan masuk mode Telnet 192.168.88.12 kemudian mengatur rfConfig, phyConfig, dan 4n lupa. Setelah mengatur parameter yang ada di SS, kemudian melakukan monitoring SS apakah pengaturan telah sesuai dengan BS. Gambar 4.10 memperlihatkan status terhubungnya SS dan BS.



```

Telnet 192.168.101.1
SUOH# show status
Structure --- <<SS Mngt Status Data>>

Downlink General
FpCount:.....90787
FpErrCount:.....0
FpCrcCount:.....1
MapCount:.....3631
MapErrCount:.....0
MapCrcCount:.....0
ByteCount:.....0
SduCount:.....0
MpdCount:.....7
HCrcErrCount:.....0
CrcErrCount:.....0

Uplink General
MapCount:.....90787
MapErrCount:.....0
MapCrcCount:.....0
ByteCount:.....38643
SduCount:.....355
MpdCount:.....362

Management
RxCount:.....34
ErrCount:.....0
CrcCount:.....0

Downlink Chan Desc
RxCount:.....454
ErrCount:.....0
CrcCount:.....0
ChangeCount:.....0

Uplink Chan Desc
RxCount:.....453
ErrCount:.....0
CrcCount:.....0

Others
Rssi:.....-16.2
Snr:.....28.5
FreqOffset:.....-1
Tcnt:.....4294967295
ModemResets:.....2
LostFrames:.....1
FrameDuration:.....10000
RngTimeCorrection:.....-101
TotalHCrcErrors:.....0
TotalCrcErrors:.....0
TotalTxBurstCount:.....6557
TotalRngReqCount:.....5
TotalBoReqCount:.....335
TotalHgmSentCount:.....33
TotalPaddingCount:.....265
RFRssi:.....-60.3
TxPower:.....1.5
LinkStatus:.....Registered(1)

SUOH#

```

Gambar 4.10 Monitoring Subscriber Station

(Sumber: Penelitian)

4.3.3 Konfigurasi Switch dan Client

Switch adalah perangkat yang berfungsi untuk menghubungkan antara SS dan Client pada penelitian kali ini, sehingga sebuah SS dapat diakses oleh 4 Client sekaligus sebagaimana tertera pada skenario penelitian ini.



Gambar 4.11 Perangkat Switch TP-Link TL-SL2210

(Sumer: Perancangan)

Perangkat yang yang digunakan adalah TP-Link TL-SL2210 yang memiliki 8 port 10/100 Mbps RJ45, 1 port 10/100/1000 Mbps RJ45, dan sebuah Gigabit SFP Slot. Namun yang digunakan hanya 5 port 10/100 Mbps RJ45 dengan rincian 4 port terhubung pada client

dan 1 port terubung pada SS. Gambar dibawah ini menunjukkan konfigurasi keseluruhan dari BS, SS, Switch, dan Client.



Gambar 4.12 Gambaran Percobaan

(Sumber: Perancangan)

4.4 Pengujian Koneksi Jaringan

Pengujian koneksi jaringan ini bertujuan untuk mengetahui koneksi antara Server, BS, SS, Switch, Client (Laptop) telah terhubung dengan baik. Pengujian dilakukan dengan cara melakukan perintah “ping” pada command prompt dari laptop menuju server. IP Server adalah 192.168.88.30, sehingga perintah ping ditunjukkan pada alamat IP tersebut.

```

Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\LAFUAN>ping 192.168.88.30

Pinging 192.168.88.30 with 32 bytes of data:
Reply from 192.168.88.30: bytes=32 time=45ms TTL=128
Reply from 192.168.88.30: bytes=32 time=23ms TTL=128
Reply from 192.168.88.30: bytes=32 time=22ms TTL=128
Reply from 192.168.88.30: bytes=32 time=21ms TTL=128

Ping statistics for 192.168.88.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 21ms, Maximum = 45ms, Average = 27ms

C:\Users\LAFUAN>
  
```

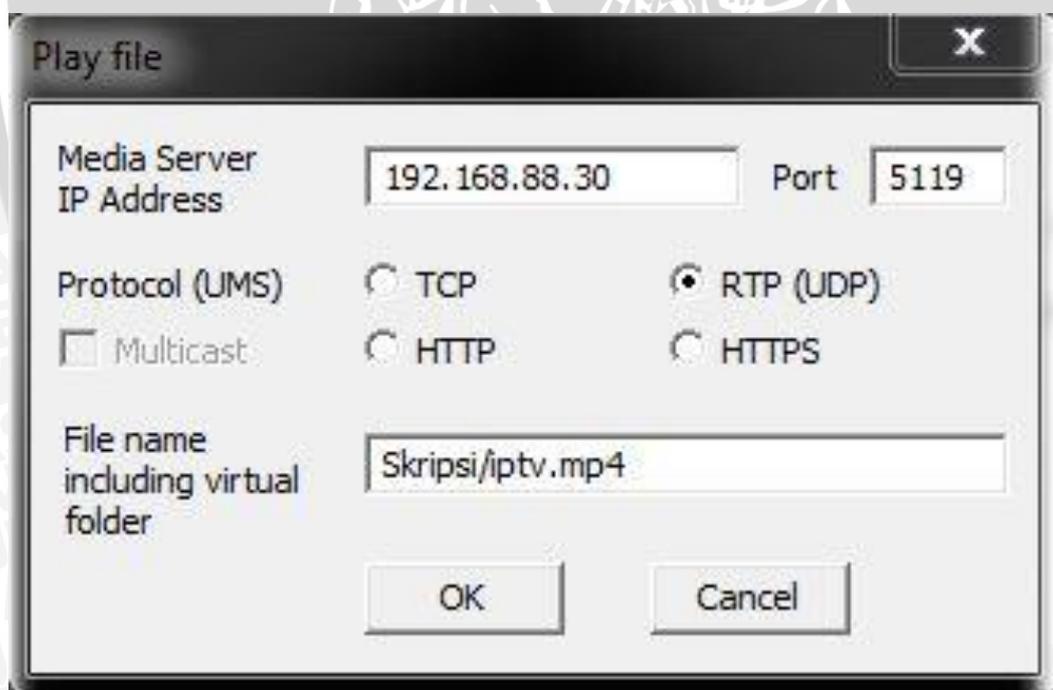
Gambar 4.13 Hasil Pengujian Koneksi Jaringan

(Sumber: Hasil Pengujian)

Perintah “ping 192.168.88.30” menunjukkan alamat server yang berada di Laboratorium Komputasi dan Jaringan yang digunakan pada penelitian kali ini untuk menguji bahwa perangkat WiMAX telah terhubung dengan baik. Kemudian informasi reply from 192.168.88.30 merupakan data pengamatan yang digunakan sebagai parameter dasar pengujian suatu jaringan. Kalimat “bytes=32” menunjukkan jumlah data yang dikirimkan pada saat pengujian, “Time=21ms mengidentifikasi respon waktu yang diperlukan dari alamat IP yang dituju menuju client kembali, dan “TTL=128” merupakan time-to-live menunjukkan jumlah maksimal melewati IP yaitu sebanyak 128 kali.

4.5 Pengujian Akses IPTV

Pengujian akses IPTV ini dilakukan untuk mengetahui apakah client sudah terhubung dengan server dengan baik, dengan cara mengakses layanan IPTV menggunakan aplikasi Unreal Streaming Media Player dengan memasukkan alamat IP server dan port yang digunakan. Dengan meakses IP server yang telah ditentukan maka *client* akan menerima siaran IPTV dengan baik berikut merupakan tampilan yang diperlihatkan ketika *client* telah dapat mengakses jaringan IPTV



Gambar 4.14 Cara Akses IPTV

(Sumber: Pengujian)



Gambar 4.15 Hasil Pengujian Koneksi IPTV

(Sumber: Pengujian)

Gambar 4.14 Menunjukkan cara akses IPTV melalui aplikasi yang telah disebutkan sebelumnya, dan pada gambar 4.15 menampilkan layanan IPTV dengan kualitas gambar yang baik menunjukkan bahwa *client* sudah dapat mengakses layanan tersebut dengan baik dan benar. Setelah melakukan pengecekan koneksi dari jaringan IPTV ke *client* maka pengukuran parameter yang dibutuhkan seperti *delay*, *packet loss*, dan *throughput* sudah dapat dilakukan.

4.6 Hasil Pengujian dan Hasil Perhitungan

Pengujian layanan IPTV pada WiMAX ditampilkan pada aplikasi Unreal Streaming Media Player, metode pengujian dilakukan dengan memvariasikan berapa banyak yang mengakses layanan tersebut dimulai dengan 1 hingga 4 client sekaligus. Sesuai dengan tujuan penelitian kali ini parameter kelas layanan yang mulai dari *real Time Polling Service* (rTPS), *non real Time Polling Service* (nrTPS), *bE* (best Effort), dan *unsolicited Grant Service* (uGS).

Proses pengambilan data menggunakan aplikasi Wireshark yang terletak pada sisi *user*. Wireshark menangkap, membaca dan menganalisis aliran data yang melewati *interface*

user. Interface pada *user* melalui jaringan kabel (*adapter* RJ-45). Proses *capturing* paket data dilakukan selama 60 detik. Paket data yang tertangkap oleh *wireshark* kemudian disimpan menjadi *file* dengan ekstensi *media library* pcap (*.pcap). Data yang didapatkan kemudian dianalisis (didekodekan) menjadi aliran RTP. Ketika data sudah menjadi aliran RTP, maka data dapat diolah menjadi parameter-parameter yang dibutuhkan, yakni *delay*, *packet loss* dan *throughput*. Luaran dari analisis data adalah *file* *wireshark* pcap yang telah didekodekan sebagai protokol RTP.

Pengolahan data dilakukan setelah pengambilan dan analisis data. Pengolahan data menghasilkan parameter-parameter QoS (*delay*, *packet loss*, *throughput*) dan jumlah paket pada setiap konfigurasi ditampilkan pada sub-bab berikutnya.

4.6.1 Hasil Melalui Pengujian

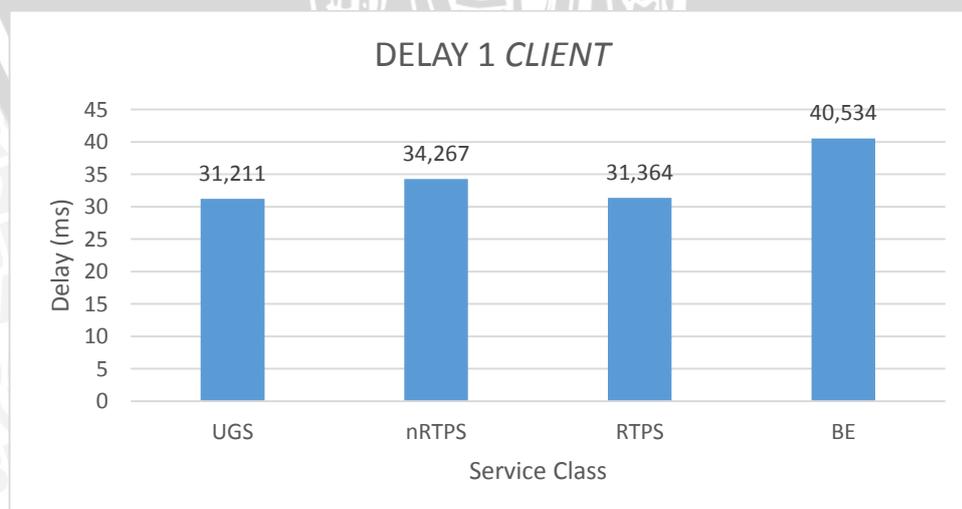
Hasil *delay*, *packet loss*, dan *throughput* yang ditampilkan pada perangkat lunak *WireShark* akan dihasilkan setelah data melalui proses pengolahan data.

4.6.1.1 Delay

Delay menunjukkan besarnya waktu tunda mulai dari *server* yang terletak di Laboratorium Komputasi dan Jaringan Komputer, Fakultas Teknik Universitas Brawijaya hingga *user* yang terletak di Laboratorium Telekomunikasi menggunakan aplikasi *remote desktop connection*.

- Skenario Akses 1 *Client*

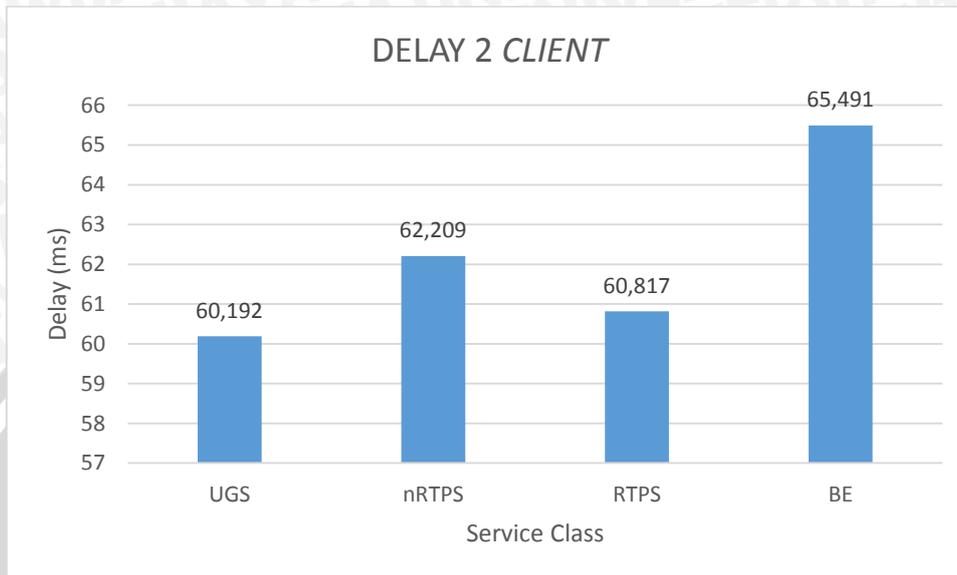
Dilihat dari Gambar 4.17 bahwa nilai *delay* BE cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan UGS. Dan nilai UGS mendapatkan nilai terendah pada angka 31,211 ms.



Gambar 4.16 Pengaruh variasi service class pada delay dengan akses 1 *Client*
(Sumber: Hasil Pengujian)

- Skenario Akses 2 *Client*

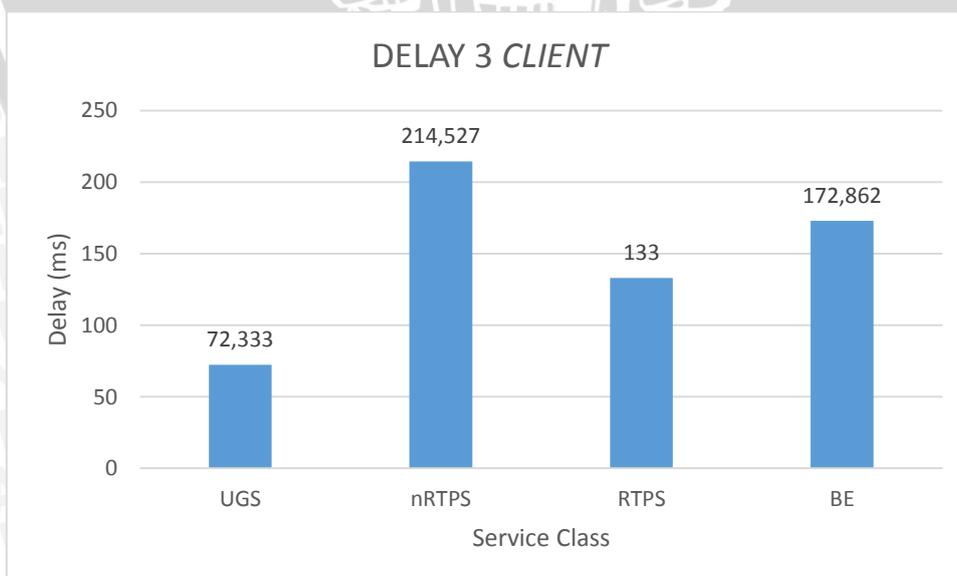
Dilihat dari Gambar 4.18 bahwa nilai delay BE cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan UGS. Dan nilai UGS mendapatkan nilai terendah pada angka 60,192 ms.



Gambar 4.17 Pengaruh variasi service class pada delay dengan akses 2 *Client*
(Sumber: Hasil Pengujian)

- Skenario Akses 3 *Client*

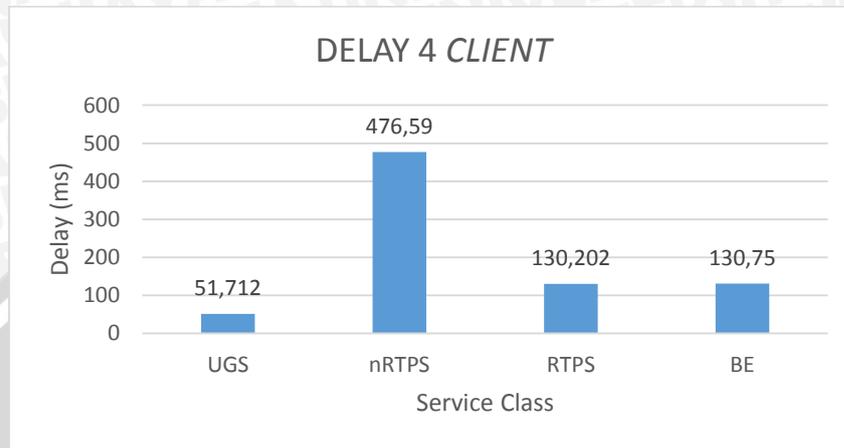
Dilihat dari Gambar 4.19 bahwa nilai delay nRTPS cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan UGS. Dan nilai UGS mendapatkan nilai terendah pada angka 72,333 ms.



Gambar 4.18 Pengaruh variasi service class pada delay dengan akses 3 *Client*
(Sumber: Hasil Pengujian)

- Skenario Akses 4 *Client*

Dilihat dari Gambar 4.20 bahwa nilai delay BE cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan UGS. Dan nilai UGS mendapatkan nilai terendah pada angka 60,192 ms.



Gambar 4.19 Pengaruh variasi service class pada delay dengan akses 4 *Client*
(Sumber: Hasil Pengujian)

Pada tabel 1 diperlihatkan nilai *delay* rata rata dengan variasi *service class* dimana nilai yang paling kecil dihasilkan dari *service class* UGS dengan nilai saat menggunakan 1 *Client* adalah 31,211 ms, 2 *Client* 60,192 ms, 3 *Client* 72,33 ms, dan 4 *Client* 51,712 ms. Sedangkan nilai delay rata rata terbesar dihasilkan dari menggunakan service class BE dan nRTPS dengan nilai saat menggunakan 1 *Client* adalah 40,534 ms dan 40,534 ms, 2 *Client* 62,209 ms dan 65,491ms, 3 *Client* 214,527 ms dan 172,862 ms, dan 4 *Client* 476,59 ms dan 130,75 ms.

Tabel 4.1 *Delay* rata rata dengan variasi service class

NO	SERVICE CLASS	DELAY (ms)			
		1 Client	2 Client	3 Client	4 Client
1	UGS	31,211	60,192	72,333	51,712
2	nRTPS	34,267	62,209	214,527	476,59
3	RTPS	31,364	60,817	133	130,202
4	BE	40,534	65,491	172,862	130,75

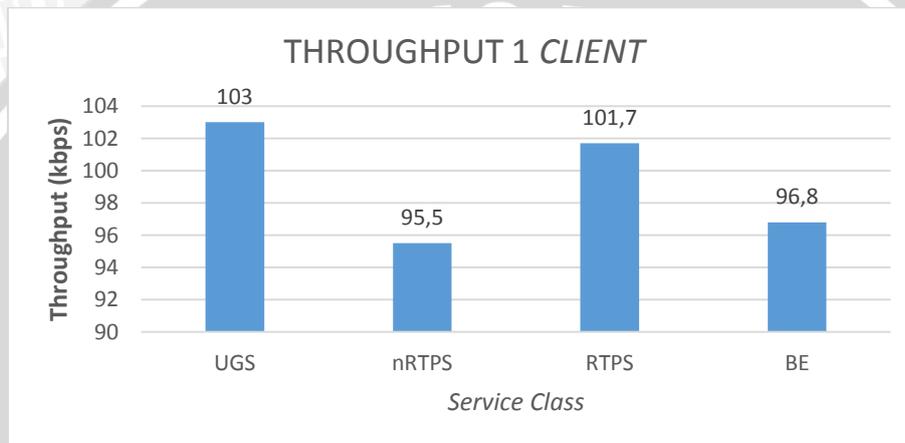
(Sumber: Hasil Pengujian)

4.6.1.2 Throughput

Throughput menunjukkan kecepatan penerimaan paket data yang diterima pengguna dengan keadaan benar terhadap waktu transmisi yang dibutuhkan dari sumber data pengirim ke penerima. *Throughput* memiliki satuan *bit/detik*.

- Skenario Akses 1 *Client*

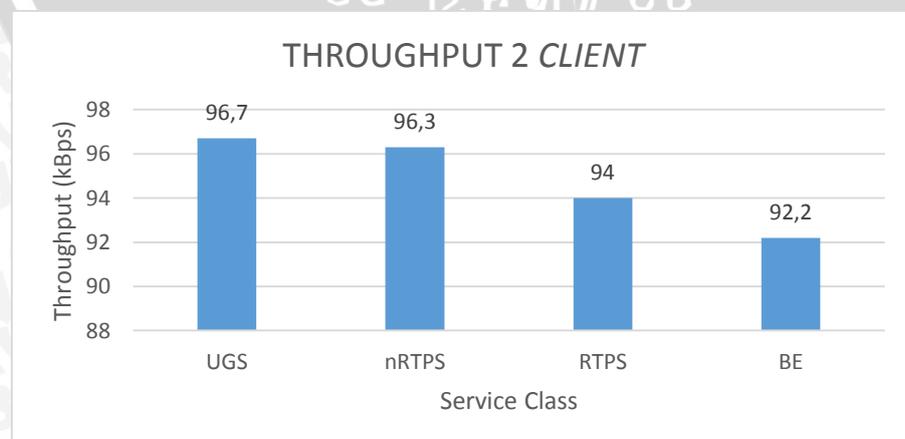
Dilihat dari Gambar 6. bahwa nilai delay UGS dan RTPS cenderung lebih tinggi dibandingkan nilai dari nRTPS dan BE. Dan nilai UGS mendapatkan nilai tertinggi pada angka 103 kbps.



Gambar 4.20 Pengaruh variasi service class pada troughput dengan akses 1 *Client* (Sumber: Hasil Pengujian)

- Skenario Akses 2 *Client*

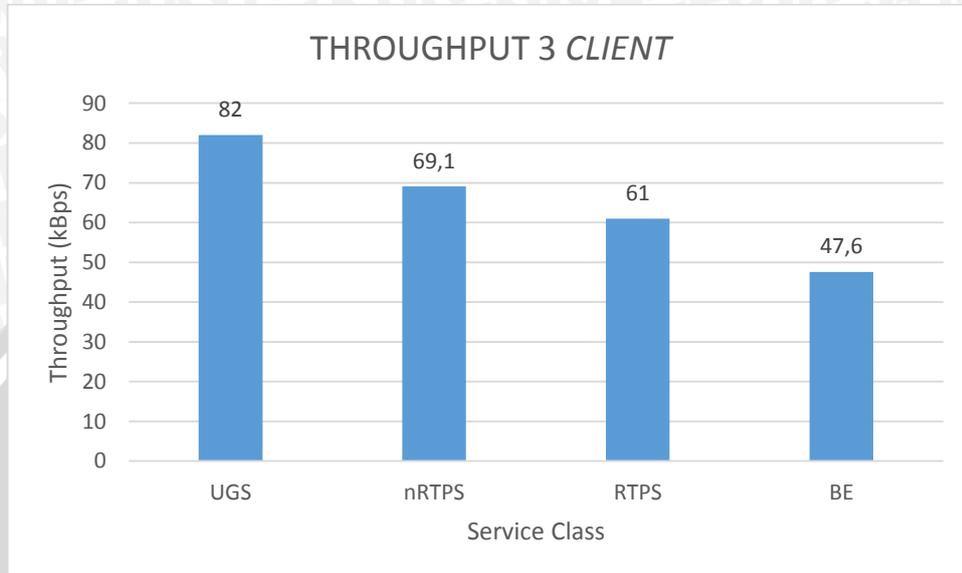
Dilihat dari Gambar 7. bahwa nilai delay UGS cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan BE. Dan nilai UGS mendapatkan nilai tertinggi pada angka 96,7 kbps.



Gambar 4.22 Pengaruh variasi service class pada troughput dengan akses 2 *Client* (Sumber: Hasil Pengujian)

- Skenario Akses 3 *Client*

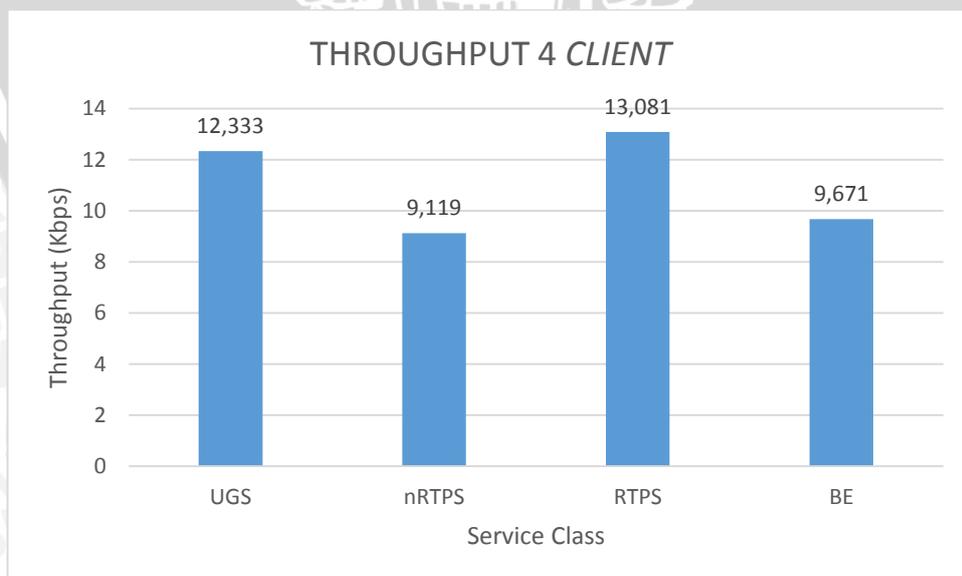
Dilihat dari Gambar 4.23 bahwa nilai delay UGS cenderung lebih tinggi dibandingkan nilai dari nRTPS, RTPS, dan BE. Dan nilai UGS mendapatkan nilai tertinggi pada angka 82 kBps.



Gambar 4.21 Pengaruh variasi service class pada troughput dengan akses 2 *Client* (Sumber: Hasil Pengujian)

- Skenario Akses 4 *Client*

Dilihat dari Gambar 4.23 bahwa nilai delay UGS dan RTPS cenderung lebih tinggi dibandingkan nilai dari nRTPS, dan BE. Dan nilai RTPS mendapatkan nilai tertinggi pada angka 13,081 kBps.



Gambar 4.22 Pengaruh variasi service class pada troughput dengan akses 4 *Client* (Sumber: Hasil Pengujian)

Pada tabel 4.23 diperlihatkan nilai Throughput rata rata dengan variasi *service class* dimana nilai yang paling besar dihasilkan dari *service class* UGS dengan nilai saat menggunakan 1 *Client* adalah 103 kbps, 2 *Client* 96,7 kbps, 3 *Client* 82 kbps, 4 *Client* 12,333 kbps . Sedangkan nilai delay rata rata terendah dihasilkan dari menggunakan *service class* BE dengan menggunakan 1 *Client* adalah 96,8 kbps, 2 *Client* 92,2 kbps, 3 *Client* 47,6 kbps, 4 *Client* 9,671 kbps.

Tabel 4.2 *Throughput* rata rata dengan variasi *service class*

NO	SERVICE CLASS	Throughput (kbps)			
		1 Client	2 Client	3 Client	4 Client
1	UGS	103	96,7	82	12,333
2	nRTPS	95,5	96,3	69,1	9,119
3	RTPS	101,7	94	61	13,081
4	BE	96,8	92,2	47,6	9,671

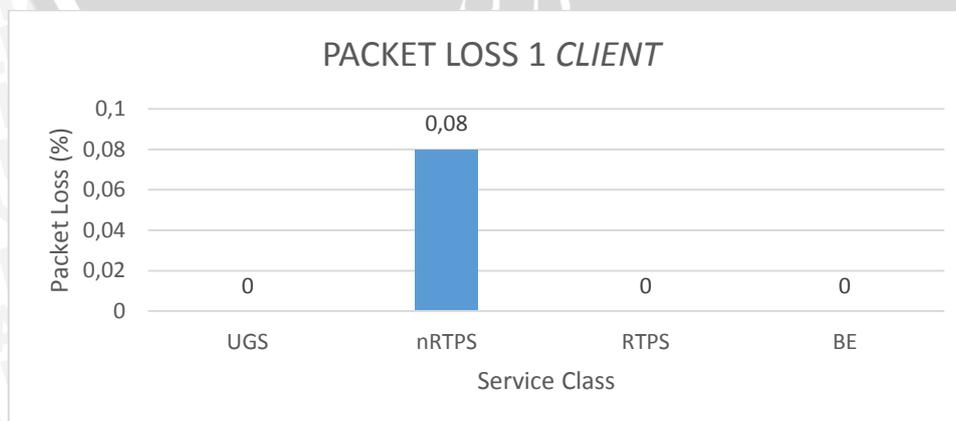
(Sumber: Hasil Pengujian)

4.6.1.3 Packet Loss

Packet Loss pada penelitian menunjukkan nilai rasio dalam persen paket yang hilang dengan jumlah paket keseluruhan yang dikirimkan dari BS ke SS.

- Skenario Akses 1 *Client*

Dilihat dari Gambar 4.24 bahwa nilai delay nRTPS cenderung lebih tinggi dibandingkan nilai dari UGS, RTPS, dan BE. Dan nilai nRTPS mendapatkan nilai tertinggi pada angka 0,08 %.

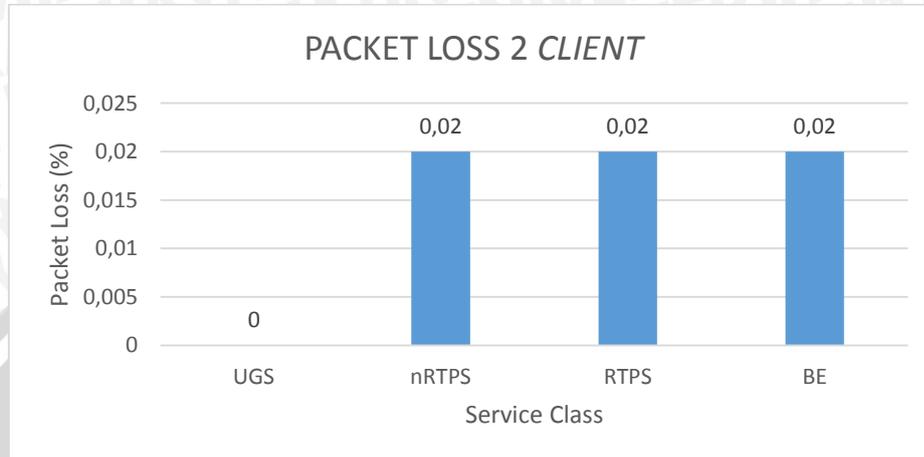


Gambar 4.23 Pengaruh variasi *service class* pada *packet loss* dengan akses 1 *Client*

(Sumber: Hasil Pengujian)

- Skenario Akses 2 *Client*

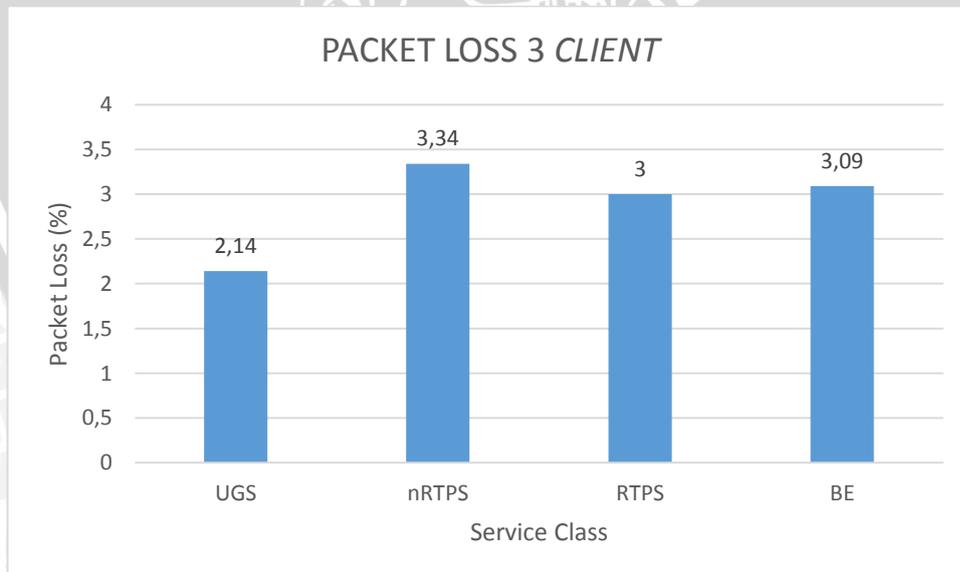
Dilihat dari Gambar 4.25 bahwa nilai delay UGS cenderung lebih rendah dibandingkan nilai dari nRTPS, RTPS, dan BE. Dan nilai nRTPS, RTPS, dan BE mendapatkan nilai tertinggi pada angka 0,02 %.



Gambar 4.24 Pengaruh variasi service class pada *packet loss* dengan akses 1 *Client* (Sumber: Hasil Pengujian)

- Skenario Akses 3 *Client*

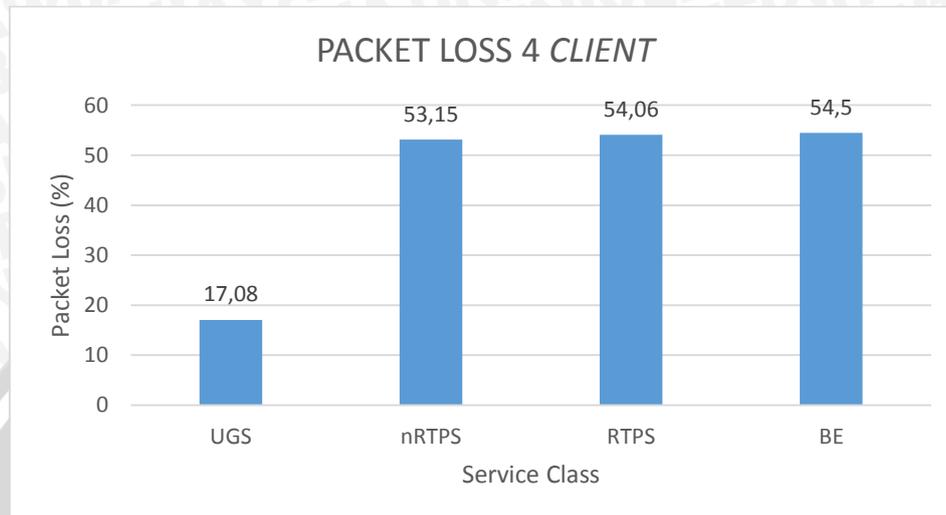
Dilihat dari Gambar 4.26 bahwa nilai delay UGS cenderung lebih rendah dibandingkan nilai dari nRTPS, RTPS, dan BE. Dan nilai UGS mendapatkan nilai tertinggi pada angka 2,14 %.



Gambar 4.25 Pengaruh variasi service class pada *packet loss* dengan akses 3 *Client* (Sumber: Hasil Pengujian)

- Skenario Akses 4 *Client*

Dilihat dari Gambar 4.27 bahwa nilai delay UGS cenderung lebih rendah dibandingkan nilai dari nRTPS, RTPS, dan BE. Dan nilai UGS mendapatkan nilai tertinggi pada angka 17,8 %.



Gambar 4.26 Pengaruh variasi service class pada *packet loss* dengan akses 4 *Client* (Sumber: Hasil Pengujian)

Pada tabel 4.3. diperlihatkan nilai Packet Loss rata rata dengan variasi *service class* dimana nilai yang paling rendah dihasilkan dari *service class* UGS dengan nilai saat menggunakan 1 *Client* adalah 0 %, 2 *Client* 0 %, 3 *Client* 2,14 %, 4 *Client* 17,08 %. Sedangkan nilai packet loss rata rata terendah dihasilkan dari menggunakan service class BE dan nRTPS.

Tabel 4.3 *Packet Loss* rata rata dengan variasi service class

NO	SERVICE CLASS	Packet Loss (%)			
		1 Client	2 Client	3 Client	4 Client
1	UGS	0	0	2,14	17,08
2	nRTPS	0,08	0,02	3,34	53,15
3	RTPS	0	0,02	3	54,06
4	BE	0	0,02	3,09	54,5

(Sumber: Hasil Pengujian)

4.6.2 Hasil Melalui Perhitungan Teoritis

Pada bagian ini dihasilkan parameter jaringan yaitu *delay end-to-end*, *packet loss* dan *throughput* melalui perhitungan dengan literatur yang ada, standar perangkat atau standar WiMAX IEEE 802.16d.

4.6.2.1 Spesifikasi IPTV

Spesifikasi paket *video conference* yang digunakan adalah sebagai berikut:

1. Pada penelitian ini menggunakan 2 jenis *codec*, yaitu *codec* H.264 untuk video dan *codec* AAC LC untuk audio, spesifikasi masing-masing *codec* dijelaskan dibawah:
 - *Audio codec* adalah AAC LC
Menggunakan *bitrate codec* 125 Kbps, *delay codec* 54,6 ms dan *payload maksimum* 63 byte
 - *Video codec* adalah H.264/MPEG-4 AVC
Menggunakan *bitrate* 548 Kbps, *delay codec* 50 ms dan *payload maksimum* 254 byte
2. *Frame rate* IPTV yang dipergunakan adalah 25 ms
3. Protokol: RTP, UDP dan IP
4. Panjang *header* NALU = 1 byte
5. Panjang *header* RTP = 12 byte
6. Panjang *header* IP = 20 byte
7. Panjang *header* UDP = 8 byte
8. Probabilitas *bit error* = 10^{-7}

Untuk menghitung besar *payload video* dan *payload audio* pada layanan IPTV tiap *frame* ditentukan persamaan 2.1 dan 2.2, untuk video 480p,

$$P_{LA480p} = \text{Bitrate}_{\text{audio}} \times \text{frame rate}$$

$$= (125 \cdot 10^3 \text{ bps}) \times (25 \cdot 10^3) = 3125 \text{ bit}$$

$$P_{LV480p} = \text{Bitrate}_{\text{video}} \times \text{frame rate}$$

$$= (548 \cdot 10^3 \text{ bps}) \times (25 \cdot 10^3) = 13700 \text{ bit}$$

Payload video dan *audio* kemudian dienkodkan menggunakan *codec*. *Payload video* dienkodkan H.264/ AVC MPEG-4 dan *payload audio* dienkodkan AAC LC. Jumlah *payload* disegmentasi berdasarkan *payload* yang ditentukan oleh *codec*. Sehingga jumlah nilai paket untuk *audio* dan *video* dapat diperoleh dengan menggunakan persamaan (2-3) dan (2-4):

$$N_A = \frac{P_{LA}}{P_{LA \max}} = \frac{3125}{63 \times 8} = 6,20 \approx 6 \text{ paket}$$

$$N_V = \frac{P_{LV}}{P_{LV \max}} = \frac{13700}{254 \times 8} = 6,74 \approx 7 \text{ paket}$$

Kemudian jumlah *payload audio* dan *video* ditambah dengan *header* NALU, RTP, UDP dan IP menggunakan persamaan (2-5) dan (2-6) adalah:

$$\begin{aligned}
 \text{Paket data audio} &= P_{LA} + (N_A \times (NALU + H_{RTP} + H_{UDP} + H_{IP})) \\
 &= 3125 + (6 \times (8 + 96 + 64 + 160)) \text{ bit} \\
 &= 5093 \text{ bit}
 \end{aligned}$$

$$\begin{aligned}
 \text{Paket data video} &= P_{LV} + (N_V \times (NALU + H_{RTP} + H_{UDP} + H_{IP})) \\
 &= 13700 + (7 \times (8 + 96 + 64 + 160)) \text{ bit} \\
 &= 15996 \text{ bit}
 \end{aligned}$$

Sehingga jumlah besar paket data aplikasi *video conference* yang ditransmisikan pada sistem ditentukan menggunakan persamaan (2-7) didapat:

$$W_{data} = P_{data\ audio} + P_{data\ video} = (5093 + 15996) \text{ bit} = 21089 \text{ bit}$$

$$W_{data} = 2636 \text{ byte}$$

4.6.2.2 Delay End-to-End

$$t_{end\ to\ end} = t_{codec} + t_{MAN}$$

- Analisis Delay MAN

$$t_{MAN} = t_{proses} + t_{prop} + t_{trans} + t_w$$

Delay Proses (t_{proses})

Pada layer transportasi paket data *audio* dan data *video* mengalami perubahan *format* dengan adanya penambahan *header* RTP dan UDP sehingga format data menjadi segmen, kemudian besar data ditambah *header* RTP (persamaan 2-11):

$$\begin{aligned}
 W_{message} &= W_{data} + \text{Header}_{RTP} \\
 &= (2636 + 12) \text{ byte} \\
 &= 2648 \text{ byte}
 \end{aligned}$$

$$\begin{aligned}
 W_{segmen} &= W_{message} + \text{Header}_{UDP} \\
 &= 2648 + 8 \\
 &= 2656
 \end{aligned}$$

$$\begin{aligned}
 W_{datagram} &= W_{segmen} + \text{Header}_{IP} \\
 &= 2656 + 20 \\
 &= 2676
 \end{aligned}$$

$$\begin{aligned}
 W_{frame} &= W_{datagram} + \text{Header}_{ethernet} \\
 &= 2676 + 18 \\
 &= 2694 \text{ byte}
 \end{aligned}$$

Dari *Ethernet*, *frame Ethernet* dikirimkan ke *layer MAC WiMAX* dan mengalami penambahan *header MAC*. Berdasarkan pernyataan tersebut maka jumlah W_{frame} total sebagai berikut:

$$\begin{aligned} W_{frame\ total} &= W_{frame} + Header_{MAC} \\ &= 2687 + 6 \\ &= 2693\ byte \end{aligned}$$

Besar *delay dekapsulasi* dengan besar *delay enkapsulasi* dengan kecepatan *layer Ethernet* adalah 100 Mbps dan untuk kapasitas kanal *WiMAX Redline AN 100U* adalah 2 Mbps. Maka *delay Enkapsulasi*:

$$\begin{aligned} t_{enc1} &= \frac{W_{frame\ total}}{C_{ethernet}} \times 8 = \frac{2693}{100 \times 10^6} \times 8 = 0.00021544\ s \\ t_{enc2} &= \frac{W_{frame\ total}}{C_{WiMAX}} \times 8 = \frac{2693}{2 \times 10^6} \times 8 = 0.010772\ s \end{aligned}$$

Setelah didapat nilai dari *delay enkapsulasi*, maka nilai *delay dekapsulasi* sebagai berikut:

$$\begin{aligned} t_{dec1} &= \frac{W_{frame\ total}}{C_{ethernet}} \times 8 = \frac{2693}{100 \times 10^6} \times 8 = 0.00021544\ s \\ t_{dec2} &= \frac{W_{frame\ total}}{C_{WiMAX}} \times 8 = \frac{2693}{2 \times 10^6} \times 8 = 0.010772\ s \end{aligned}$$

Dari hasil perhitungan *delay enkapsulasi* dan *delay dekapsulasi* tersebut, maka akan didapatkan nilai *delay proses* dengan persamaan 2-24 sebagai berikut:

$$\begin{aligned} t_{proses} &= t_{enc1} + t_{enc2} + t_{dec1} + t_{dec2} \\ &= 0.00021544\ s + 0.010772\ s + 0.00021544\ s + 0.010772\ s \\ &= 0.02197488\ s \end{aligned}$$

Delay Propagasi (t_{prop})

$$t_{prop} = \frac{d_{max}}{v} = \frac{10}{3 \times 10^8} = 3,33 \times 10^{-8}\ s$$

Jadi nilai *delay Propagasi* adalah sebesar $3,33 \times 10^{-8}\ s$

Delay Transmisi (t_{trans})

$$t_{trans1} = \frac{w}{C} \times 8 = \frac{2693\ byte}{100 \times 10^6\ bps} \times 8 = 0.00021544\ s$$

$$t_{trans2} = \frac{w}{C} \times 8 = \frac{2693\ byte}{2 \times 10^6\ bps} \times 8 = 0.010772\ s$$

$$\begin{aligned}
 t_{transmisi} &= t_{trans1} + t_{trans2} \\
 &= 0.00021544s + 0.010772s \\
 &= 0.01098744 s
 \end{aligned}$$

Jadi nilai delay Transmisi adalah sebesar 0.01098744 s

Delay Antrean (t_w)

$$\mu = \frac{c}{w} = \frac{2 \times 10^6 \text{ mbps}}{2693 \times 8 \text{ bit}} = 92,83 /s$$

Besarnya kecepatan kedatangan dengan faktor utilitas = 0,1

$$\lambda_w = \mu \rho = 92,83 \times 0,1 = 9,283$$

Sehingga besarnya *delay* antrean adalah

$$\begin{aligned}
 t_w &= \frac{\lambda_w}{\mu(\mu - \lambda_w)} + \frac{1}{\mu} = \frac{9,283}{92,83(92,83 - 9,283)} + \frac{1}{92,83} \\
 &= 0.0119693106874s
 \end{aligned}$$

Jadi nilai delay antrean adalah 0.012s. Dari data hasil perhitungan *delay* proses, *delay* propagasi, *delay* transmisi dan *delay* antrean maka didapatkan hasil *delay end to end* senilai 0,544 detik.

4.6.2.3 Probabilitas *Packet Loss*

Mencari nilai *payload audio* dan *payload video* melalui persamaan (2.19) dan (2.21) dengan kapasitas kanal sebesar 1 Mbps yaitu :

$$\begin{aligned}
 P_{LA} &= B_{CODEC} \times \text{frame rate} \\
 &= (125. 10^3 \text{ bps}) \times (25. 10^3) = 3125 \text{ bit} \\
 &= 3125
 \end{aligned}$$

$$\begin{aligned}
 P_{LV} &= B_{CODEC} \times \text{frame rate} \\
 &= (548. 10^3 \text{ bps}) \times (25. 10^3) = 13700 \text{ bit} \\
 &= 13700
 \end{aligned}$$

Mencari nilai probabilitas *payload* pada layanan IPTV.

$$\begin{aligned}
 \rho &= P_{size} \cdot \rho_b \\
 &= (68 + 2636) \times 8 \times (10^{-7}) \\
 &= 21632 \times 10^{-7}
 \end{aligned}$$

Untuk mengetahui probabilitas *packet loss* pada WiMAX, dapat ditentukan oleh tipe modulasi yang digunakan. Pada penelitian ini, modulasi yang digunakan adalah modulasi

64-QAM mempunyai 6 bit untuk tiap symbol maka jumlah $n=64$ dengan bandwidth kanal (B_w)=3,75 MHz maka nilai bit ratenya.

$$R = B_w \text{ kanal} \times \log_2 n$$

$$R = (3,5 \times 10^6) \times \log_2 64$$

$$R = 21 \times 10^6 \text{ bps} = 21 \text{ Mbps}$$

Nilai probabilitas packet loss dapat dicari dengan persamaan :

$$\rho_{\text{tot}} = 1 - [(1 - \rho_{\text{net}})(1 - \rho_{\text{vid}})]$$

Dengan ρ_{net} merupakan probabilitas bit eror pada WiMAX dan ρ_{vid} adalah probabilitas *payload* layanan IPTV dengan nilai 21632×10^{-7} , maka ρ_{net} dapat dicari dengan persamaan :

$$P_{\text{network}} = \frac{e^{-\frac{Eb}{No}}}{2 \sqrt{\pi \left(\frac{Eb}{No}\right)}}$$

Untuk mencari nilai $\frac{Eb}{No}$ maka terlebih dahulu mencari nilai No dan P_r , No merupakan noise level pada WiMAX dengan nilai $-135,24$ dB dan nilai P_r merupakan daya terima yang didapat oleh penerima sebesar $-22,91$ dB dengan mengacu pada nilai yang telah diatur secara manual. Eb/No dengan panjang jarak BS ke *user* 10 m dan *bandwidth* 3,5 MHz. SNR untuk modulasi 64-QAM adalah sebesar 122,33 R adalah *bit rate* dengan nilai 21 Mbps, sehingga Eb/No dapat dihitung dengan persamaan:

$$\frac{Eb}{No} = SNR - 10 \log \frac{B}{R}$$

$$\frac{Eb}{No} = 122,33 - 10 \log \frac{3,5 \times 10^6}{21 \times 10^6}$$

$$\frac{Eb}{No} = 101,98 \text{ dB}$$

Dan P_{network} bisa dihitung dengan rumus :

$$P_{\text{network}} = \frac{e^{-\frac{Eb}{No}}}{2 \sqrt{\pi \left(\frac{Eb}{No}\right)}}$$

$$P_{network} = \frac{e^{-101,98}}{2\sqrt{3,14(101,98)}}$$

$$P_{network} = 1,46 \times 10^{-46}$$

Sehingga probabilitas *packet loss* pada layanan *live streaming* pada WiMAX dengan modulasi 64-QAM dengan kapasitas kanal 2000 Kbps dapat dihitung dengan persamaan berikut.

$$\begin{aligned}\rho_{tot} &= 1 - [(1 - \rho_{net})(1 - \rho_{vid})] \\ &= 1 - [(1 - 1,46 \times 10^{-46})(1 - 21632 \times 10^{-7})] \\ &= 0.0021632\end{aligned}$$

Dan nilai prosentase *packet loss* yang didapat adalah sebagai berikut:

$$\begin{aligned}\rho_{tot}(\%) &= \rho_{tot} \times 100 \% \\ &= 0.0021632 \times 100 \% \\ &= 0,21632\%\end{aligned}$$

4.6.2.4 Throughput

Throughput dapat dihitung dengan menggunakan persamaan (2.13), yaitu :

$$\lambda = \frac{(1 - \rho)}{t_{trans}[1 + (\alpha - 1)\rho]}$$

$$\alpha = 1 + \frac{2t_{prop} + 2t_1}{t_1}$$

$$\alpha = 3 + \frac{2t_{prop}}{t_1}; \text{ dimana } t_1 = \frac{L}{C_{kanal}}$$

Sehingga didapatkan:

$$L = \text{filesize}_{IPTV} = 2013,75 \times 10^3$$

$$C_{kanal} = 2 \times 10^6$$

$$\rho_{IPTV} = 1,6918 \times 10^{-3}$$

$$t_1 = \frac{2013,75 \times 10^3 \times 8}{2 \times 10^6} = 8,055$$

$$\alpha = 3 + \frac{2 \times 3,33 \times 10^{-8}}{8,055} = 3$$

$$\lambda = \frac{(1 - 0,00216)}{0.01098744 [1 + (3 - 1)0,00216]} = 90,426 \text{ kbps}$$

Jadi nilai *Throughput* adalah sebesar 90,426 kbps.

4.6.3. Pembahasan Performansi Layanan IPTV

1. Berdasarkan nilai delay yang diperoleh dari pengambilan data kelas layanan UGS mendapatkan nilai sebesar 31,211 ms, 60,192 ms, 72,33 ms, 51,712 ms dari pengujian dengan 1 sampai 4 client secara berturut-turut sekaligus memperlihatkan hasil terbaik dibandingkan dengan 3 kelas layanan lainnya
2. Berdasarkan nilai *throughput* kelas layanan UGS pula mendapatkan nilai terbaik dengan 103 kbps, 96,7 kbps, 82 kbps, dan 12,33 kbps dari pengujian dengan 1 sampai 4 *client* secara berturut-turut. Disusul dengan kelas layanan rtPS, nrtPS, dan BE.
3. Berdasarkan nilai *packet loss* kelas layanan UGS mendapatkan nilai 0 %, 0%, 2,14%, dan 17,08% dari pengujian dengan 1 sampai 4 *client* secara berturut-turut. Kelas layanan rtPS 0%, 0,02%, 3%, dan 54,06 % dari pengujian dengan 1 sampai 4 *client* secara berturut-turut. UGS dan rtPS mendapatkan nilai yang lebih baik dibandingkan dengan nrtPS dan BE.

