



UNIVERSITAS INDONESIA

**ANALISA LAYANAN VIDEO ON DEMAND
PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM**

SKRIPSI

**DEOLENS
0606073852**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**



UNIVERSITAS INDONESIA

**ANALISA LAYANAN VIDEO ON DEMAND
PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**DEOLENS
0606073852**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**

i

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : DEOLENS
NPM : 0606073852
Tanda Tangan :
Tanggal : 28 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Deolens
NPM : 0606073852
Program Studi : Teknik Elektro
Judul Skripsi : Analisa Layanan Video on Demand Pada
Arsitektur IP Multimedia Subsystem

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Ir. Djamhari Sirat M.Sc, Ph.D

Penguji : Dr. Ir. Muhammad Asvial M.Eng

Penguji : Prima Dewi Purnamasari ST., MT., MSc

Ditetapkan di : Depok

Tanggal : 28 Juni 2010



KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas kehendak dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa tanpa bantuan banyak pihak, skripsi ini tidak mungkin terselesaikan. Oleh sebab itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada pihak-pihak yang membantu saya dalam segala hal mengenai penyusunan skripsi ini yaitu:

- 1) Ir. Djamhari Sirat M.Sc, Ph.D selaku dosen pembimbing seminar dan skripsi saya.
- 2) Pak Noor Iza (Depkominfo), Pak Yulianus (Indosat), Pak Bambang Soekanto (Telkom), dan Pak Panca (Telkomsel) yang berhasil memberikan arahan secara umum dan teknis terhadap kebutuhan dunia telekomunikasi sekarang dan di masa yang akan datang.
- 3) Rekan-rekan satu bimbingan saya: Mark Swapo, R. Kharisma, Hartono, dan Zaneta Pelangi yang telah melewati “hari-hari sulit” untuk membangun jaringan implementasi Open IMS Core bersama-sama.
- 4) Keluarga saya yang telah banyak memberikan semangat dan dorongan dalam pembuatan skripsi ini.
- 5) Seluruh pihak di Teknik Elektro FTUI terutama untuk asisten Lab Telkom dan STL yang telah mengizinkan kami menggunakan laboratoriumnya sebagai tempat mengerjakan skripsi kami.
- 6) Teman-teman dan sahabat saya yang telah memberikan motivasi dan semangat pada saat saya mulai putus asa.

Depok, Juni 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Deolens
NPM : 0606073852
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (Non-exclusive RoyaltyFree Right) atas karya ilmiah saya yang berjudul :

ANALISA LAYANAN VIDEO ON DEMAND
PADA ARSITEKTUR IP MULTIMEDIA SUBSYSTEM

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 28 Juni 2010

Yang Menyatakan

(Deolens)

ABSTRAK

Nama : Deolens
Program Studi : Teknik Elektro
Judul : ANALISA LAYANAN VIDEO ON DEMAND PADA
ARSITEKTUR IP MULTIMEDIA SUBSYSTEM

IMS adalah *session-controll subsystem* yang didasari pada *Internet Protocol (IP)*, *Session Initiation Protocol (SIP)*, *Session Description Protocol (SDP)* dan beberapa protokol lainnya yang didesain secara khusus untuk men-support layanan-layanan multimedia yang melalui berbagai macam *access networks*.

Video on Demand adalah suatu sistem yang memungkinkan untuk memilih dan menonton/mendengar video sesuai dengan permintaan pengguna. *Video on Demand* mengkombinasikan kualitas layanan dari TV kabel dan kemampuan interaksi dari VCR.

Pada tugas akhir ini akan dirancang dan dianalisis layanan Video on Demand pada jaringan IMS. Jaringan ini akan menggunakan 2 buah *router*. Dari kedua *router*, *bandwidth* dikontrol sehingga besar *bandwidth* pada hubungan keduanya dapat dijaga. Dari implementasi ini selanjutnya akan dianalisa *bandwidth* optimum untuk layanan VoD ini sesuai dengan standart ITU-T G.1010. Dari pengujian dan analysis didapat *bandwidth* optimum untuk layanan VoD ini adalah 1536 kbps.

Kata Kunci:

bandwidth, IMS, IP, router, SDP, SIP, Video on Demand

ABSTRACT

Name : Deolens
Study Program : Teknik Elektro
Title : ANALYSIS VIDEO ON DEMAND SERVICE ON IP
MULTIMEDIA SUBSYSTEM ARCHITECTURE

IMS is session-controll subsystem that based on Internet Protocol (IP), Session Initiation Protocol (SIP), Session Description Protocol (SDP) and same other protocol which is desain for supporting multimedia services through many network access.

Video of On Demand is a conducive system to choose and watching/hearing video as according to request of consumer. Video of On Demand combine the quality of service of TV cable and ability of interaction of VCR.

In this final task will be designed and analysed Video on Demand service at IMS network. This Network will use 2 router. From both router the bandwidth is controlled so the volume of the bandwidth can be maintained. From this implementation will be analysed the optimum bandwidth for this Vod service as according to ITU-T G.1010 standard. From the result of test and analysis got the optimum bandwidth for this VoD service is 1536 kbps.

Keywords:

bandwidth, IMS, IP, router, SDP, SIP, Video on Demand

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI.....	v
ABSTRAK	vi
ABSTRACT.....	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah.....	3
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	4
2. LANDASAN TEORI	6
2.1 IP Multimedia Subsystem.....	6
2.1.1 Home Subscriber Server (HSS).....	8
2.1.2 Proxy Call Session Control Function (P-CSCF).....	8
2.1.3 Interrogating Call Session Control Function (I-CSCF).....	8
2.1.4 Serving Call Session Control Function (S-CSCF)	9
2.2 Video on Demand	9
2.3 Protocol-Protocol Pada Layanan Multimedia.....	10
2.3.1 Real Time Transport Protocol (RTP)	10
2.3.2 Real Time Control Protocol (RTCP).....	13
2.3.3 Real Time Streaming Protocol (RTSP)	15
2.3.4 Session Initiation Protocol (SIP).....	15
2.3.5 Session Description Protocol (SDP).....	17
2.4 Parameter Quality Of Service (QOS).....	18
2.4.1 Delay	19
2.3.2 Jitter	19
2.3.3 Packet Loss	19
2.3.4 Troughput.....	20
3. PERANCANGAN VIDEO ON DEMAND BERBASIS OPEN IMS.....	21
3.1 Konfigurasi dan Instalasi Core IMS	22
3.2 Instalasi dan Konfigurasi Aplication Server	30
3.3 Instalasi dan Konfigurasi Media Server	35
3.4 Instalasi dan Konfigurasi IMS Client.....	36
3.5 Pembuatan Subscriber-Subscriber Baru	37

4. UJI COBA, PENGAMBILAN DATA, DAN ANALISIS.....	39
4.1 Skenario Uji Coba.....	39
4.2 Analisa Message Flow.....	40
4.2.1 Message Flow Pada Saat Registrasi	41
4.2.2 Message Flow Pada Saat Pemanggilan VoD	42
4.2.3 Message Flow Pada Saat Pemutusan VoD	44
4.2.4 Message Flow Pada Saat Deregistrasi	45
4.3 Analisa Bandwidth Optimum	46
5. KESIMPULAN.....	52
DAFTAR REFERENSI	53

DAFTAR TABEL

Tabel 4.1 Message Flow Registrasi	41
Tabel 4.2 Message Flow Pemanggilan VoD	42
Tabel 4.3 Message Flow Pemutusan VoD	44
Tabel 4.4 Message Flow Deregistrasi	45
Tabel 4.5 Hasil Pengambilan Data	47

DAFTAR GAMBAR

Gambar 2.1 IMS Core	7
Gambar 2.2 Enkapsulasi RTP	11
Gambar 2.3 RTP Packet Format	12
Gambar 2.4 RTCP Packet Types	13
Gambar 2.5 RTCP RR Packet Format	13
Gambar 2.6 RTCP SR Packet Format	14
Gambar 2.7 Perhitungan One Way Delay dan Round Trip Delay	14
Gambar 3.1 Arsitektur VoD	21
Gambar 3.2 Tampilan File named.conf.local localhost	24
Gambar 3.3 Tampilan File resolv.conf localhost	25
Gambar 3.4 Tampilan File hosts localhost	25
Gambar 3.5 Restart Bind	26
Gambar 3.6 Ping PCSCF	26
Gambar 3.7 Dig open-ims.test	26
Gambar 3.8 Tampilan File resolv.conf spesifik IP	28
Gambar 3.9 Tampilan File hosts spesifik IP	28
Gambar 3.10 Tampilan File open-ims.dnszone spesifik IP	29
Gambar 3.11 Tampilan FHOSS	30
Gambar 3.12 VoD Application Server	31
Gambar 3.13 VoD Trigger Point	31
Gambar 3.14 iFC	32
Gambar 3.15 Shared iFC Set	32
Gambar 3.16 Service Profile	33
Gambar 3.17 Key Falue File	34
Gambar 3.18 Tampilan Application Server Pada Terminal	34
Gambar 3.19 VLC Media Player	35
Gambar 3.20 UCT IMS Client	36
Gambar 3.21 Preview VoD	36
Gambar 3.22 IMS Subscription	37
Gambar 3.23 Private User Identity	38
Gambar 3.24 Public User Identity	38
Gambar 4.1 Topologi Jaringan	39
Gambar 4.2 Message Flow	46
Gambar 4.3 Grafik Bandwidth dan Packet Loss	47
Gambar 4.4 Grafik Bandwidth dan Delay	48
Gambar 4.5 Grafik Bandwidth dan Jitter	49
Gambar 4.6 Grafik Bandwidth dan Troughput	50

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Perkembangan teknologi dewasa ini berjalan demikian pesatnya, terutama di bidang teknologi. Persaingan operator-operator dan vendor-vendor telekomunikasi dalam menyajikan suatu telekomunikasi yang cepat, aman, dan multiguna mendorong teknologi telekomunikasi untuk terus berkembang. Mulai dari *First Generation* sampai sekarang menuju ke sebuah teknologi masa depan *Next Generation Network*.

Sudut pandang utama konsep NGN adalah layanan yang meliputi *voice*, data, multimedia dan Internet. Hal terpenting dalam suatu konsep NGN adalah konvergensi dan layanan yang berbasis IP. Untuk mewujudkan konsep NGN yang menitikberatkan pada konvergensi dan layanan yang berbasis IP ini diperlukan suatu teknologi yang dikenal dengan nama *IP Multimedia Subsystem (IMS)*. Dengan adanya IMS konvergensi dan layanan yang berbasis IP bukanlah menjadi suatu hal yang mustahil lagi.

Seperti yang telah disebutkan sebelumnya, sudut pandang utama konsep NGN adalah layanan yang meliputi *voice*, data, multimedia dan internet. Salah satu layanan atau aplikasi yang sangat menarik dan menjanjikan untuk diterapkan adalah *Video on Demand*. Beberapa tahun belakangan ini *Video on Demand* menjadi salah satu layanan yang paling banyak digunakan oleh kebanyakan orang. Sebut saja pada YouTube, komunitas video online paling populer di dunia, ratusan juta video ditonton orang setiap harinya dan ratusan ribu video diupload setiap hari. Fakta ini membuat para operator dan penyedia layanan berlomba-lomba untuk menyediakan suatu jaringan yang bisa menyediakan layanan VoD dengan kualitas yang baik. Salah satu solusinya adalah dengan menggunakan IMS sebagai dasar dari jaringan ini.

Namun, Video on Demand berbasis IMS adalah layanan yang masih baru dan masih sedikit sekali orang yang menggunakannya bahkan mungkin masih sedikit pula orang yang tahu mengenai layanan ini. Oleh karena itu masih

diperlukan penelitian mengenai Video on Demand berbasis IMS ini untuk lebih memahami cara kerjanya dan untuk mengetahui apakah layanan ini sudah layak diterapkan. Untuk itulah pada skripsi ini akan diemulasikan suatu jaringan Video on Demand yang berbasis IMS untuk memahami dan menganalisa cara kerja Video on Demand terutama mengenai *message flow* dari layanan tersebut, juga untuk melihat parameter-parameter QoS (Quality of Service) untuk mengetahui apakah layanan tersebut sudah dapat diimplementasikan atau belum.

1.2 PERUMUSAN MASALAH

Pada penulisan skripsi ini akan dibangun atau diemulasikan suatu jaringan *Video on Demand* yang berbasis IMS dengan menggunakan OpenIMSCore dan komponen-komponen lainnya yang juga menggunakan modul-modul dari OpenIMS. Dari jaringan yang dibuat ini pembahasan akan difokuskan pada *message flow* dan parameter-parameter QoS (*Quality of Service*) yang menentukan bagus atau tidaknya kualitas jaringan ini.

Berdasarkan gambaran di atas, dapat dirumuskan masalah-masalah utama yang akan dibahas pada penelitian ini, yaitu:

1. Bagaimanakah konsep jaringan IMS yang diperlukan untuk layanan *Video on Demand* ini?
2. Bagaimana *message flow* yang terjadi dari mulai registrasi, pemanggilan *Video on Demand*, pemutusan *Video on Demand*, sampai pemutusan hubungan dengan OpenIMSCore?
3. Pada bandwidth berapakah layanan *Video on Demand* berbasis IMS ini dapat bekerja secara optimum?

1.3 TUJUAN PENELITIAN

Tujuan dari pembuatan skripsi dan penelitian mengenai *Video on Demand* dengan OpenIMS ini adalah:

1. Membangun dan mengemulasikan suatu jaringan *Video on Demand* yang berbasis OpenIMS.

2. Memahami *message flow* yang terjadi dari mulai registrasi, pemanggilan *Video on Demand*, pemutusan *Video on Demand*, sampai pemutusan hubungan dengan OpenIMSCore.
3. Menganalisa pada bandwidth berapakah layanan *Video on Demand* ini dapat bekerja secara optimum.

1.4 BATASAN MASALAH

Pada penulisan skripsi ini, penelitian hanya dibatasi pada pengemuliasian layanan *Video on Demand* berbasis IMS dengan menggunakan komponen sebagai berikut:

1. Core : Open IMS Core
2. Client : UCT IMS Client
3. Media Server : VLC
4. Application Server : UCT IPTV Advanced
5. Sistem Operasi ; Ubuntu 8.04

Selain itu, penelitian ini juga hanya dibatasi pada analisa *message flow* dan pencarian *bandwidth* optimum layanan VoD ini

1.5 METODE PENELITIAN

Metode penelitian yang digunakan dalam penyusunan skripsi ini meliputi:

1. Pendekatan dari tinjauan pustaka, yaitu dengan melakukan studi literatur dari buku-buku pustaka, *website*, dan modul instalasi dari perangkat yang digunakan
2. Pendekatan dari narasumber, yaitu melalui diskusi dan tanya jawab dengan orang-orang yang ahli pada bidang-bidang yang berhubungan.
3. Perancangan perangkat lunak.
4. Pengujian dan pengambilan data.
5. Analisa Data

Penelitian dilakukan melalui prosedur sebagai berikut:

1. Instalasi dan konfigurasi pada Server:

- a) Penginstalan OpenIMSCore
 - b) Penginstalan dan konfigurasi Application Server
 - c) Penginstalan dan konfigurasi Media Server
2. Penginstalan IMS client
 3. Registrasi IMS Client pada IMS Core
 4. Pengaturan bandwidth pada router
 5. Request dan inisiasi salah satu *Video on Demand* di IMS client
 6. Pengukuran QoS jaringan IMS menggunakan Wireshark
 7. Analisa *message flow* dan QoS hasil pengukuran

1.6 SISTEMATIKA PENULISAN

Penulisan Skripsi ini dibagi menjadi beberapa bagian:

BAB 1 PENDAHULUAN

Bab ini berisikan latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini membahas mengenai teori-teori yang mendasari penelitian skripsi ini, yaitu mengenai konsep dasar IMS, *Video on Demand*, protocol-protocol yang digunakan, dan parameter-parameter QoS yang digunakan pada penelitian ini.

BAB 3 PERANCANGAN

Bab ini membahas mengenai cara instalasi dan konfigurasi IMS Core, IMS Client, Application Server, dan Media Server.

BAB 4 UJI COBA DAN ANALISIS

Bab ini berisi mengenai analisa *message flow* dari jaringan *Video on Demand* ini dan pengukuran parameter-parameter QoS pada bandwidth yang berbeda-beda.

BAB 5 KESIMPULAN

Bab ini berisi mengenai kesimpulan dari analisis yang didapat melalui penelitian ini.



BAB 2

LANDASAN TEORI

2.1. IP MULTIMEDIA SUBSYSTEM

Perkembangan teknologi dewasa ini berjalan demikian pesatnya, terutama di bidang telekomunikasi yang pada akhirnya mendorong industri telekomunikasi untuk bergerak ke arah NGN. Hal terpenting dalam suatu konsep NGN adalah konvergensi dan layanan yang berbasis IP. Untuk mewujudkan konsep NGN yang menitikberatkan pada konvergensi dan layanan yang berbasis IP ini diperlukan suatu teknologi yang dikenal dengan nama *IP Multimedia Subsystem (IMS)*.

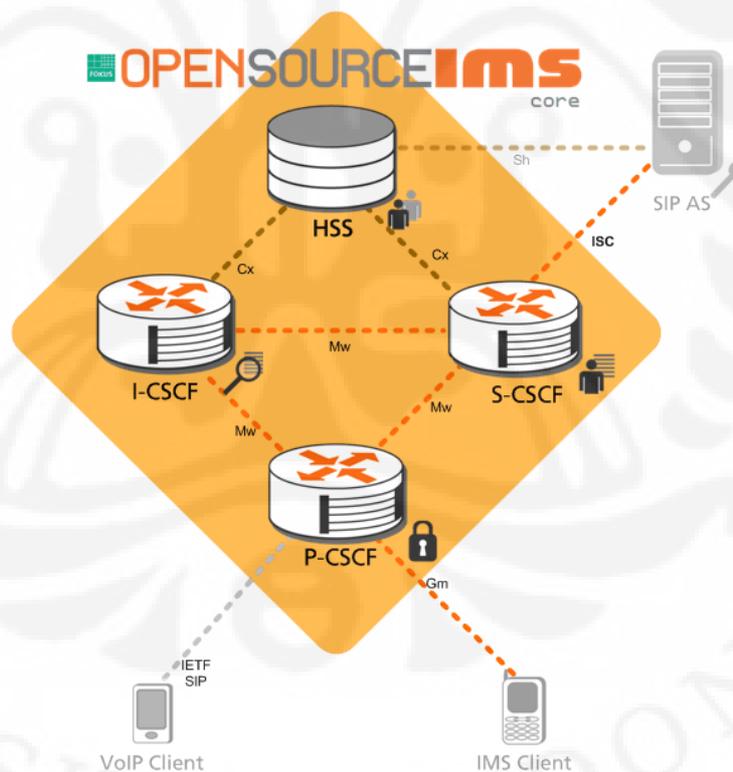
IMS adalah session-controll subsystem yang didasari pada Internet Protocol (IP), Session Initiation Protocol (SIP), Session Description Protocol (SDP) dan beberapa protocol lainnya yang didesain secara khusus untuk men-support layanan-layanan multimedia yang melalui berbagai macam access networks. Sebagai session control, IMS menyediakan beberapa fungsi seperti subscriber profile management, mekanisme charging, dan alokasi QoS pada media transmisi.

Komponen-komponen penting pada arsitektur IMS adalah Call Session Control Function (CSCF) dan Homen Subscriber Server (HSS). Dengan adanya komponen-komponen tersebut IMS dapat men-support berbagai macam access network [1], seperti:

- Wireline networks, seperti DSL, PON, Data Over Cable Service Interface Specifications (DOCSIS), dan Ethernet;
- Cellular mobile networks, seperti WCDMA, CDMA2000, GSM, dan GPRS
- IEEE wireless networks, seperti Wi-Fi dan WiMAX

Ada berbagai macam aplikasi potensial yang dapat diperoleh oleh IMS subscribers, diantaranya adalah video telephony, push to talk over cellular (PoC), instant messaging dan presence (IMP), multiparty conferencing, IPTV, game interaktif, dan masih banyak yang lainnya. Untuk menyediakan aplikasi-aplikasi ini IMS harus dapat menjalankan advance session control dengan cara yang berbeda-beda, yaitu:

- *Menyesuaikan aplikasi-aplikasi berdasarkan user equipment dan access network*, dengan fungsi ini network dapat mengurangi bitrate sinyal mengurangi bandwidth, atau mengurangi ukuran dari image di aplikasi video telephony pada saat video ditampilkan pada layar yang kecil di mobile phone.
- *Memodifikasi sesi sesuai dengan permintaan pengguna*, dengan fungsi ini IMS dapat memenuhi permintaan pelanggan yang ingin mengatur mobile phone-nya untuk berpindah dari mode Wi-Fi di kantor ke mode 3G di rumah.
- *Memungkinkan pengguna untuk mengakses layanan yang pada home network ketika berada pada visited network*, fungsi ini biasanya digunakan oleh pengguna IPTV yang ingin menonton channel yang disediakan oleh home network ketika dia sedang berada pada visited network.



Gambar 2.1 IMS Core

Sumber: <http://www.openimscore.org/>

Seperti yang telah disebutkan sebelumnya, HSS dan CSCF adalah 2 komponen terpenting dalam IMS, dimana fungsi dari masing-masing komponen tersebut adalah sebagai berikut:

2.2.1. Home Subscriber Server (HSS)

Dalam implementasinya, IMS menyediakan layanan kepada setiap pengguna sesuai dengan profilnya masing-masing. Untuk itu, IMS harus menyimpan semua identitas dan informasi dari pengguna, termasuk mengenai layanan apa saja yang diizinkan untuk mereka gunakan dan server atau network apa saja yang boleh mereka akses. Semua data-data tersebut tersimpan dalam Home Subscriber Server atau HSS.

HSS juga berfungsi untuk menyediakan informasi mengenai pengguna mana saja yang terhubung dengan network, pengguna mana saja yang sedang teregister, dan dimana lokasi tempat pelanggan tersebut berada. Untuk itu, ketika S-CSCF akan mengautentikasi pengguna dan menentukan rute signalling ke tujuan yang benar, S-CSCF harus mengakses ke HSS. Hubungan antara S-CSCF dan HSS ini dilakukan pada C_x reference point menggunakan protocol Diameter yang didesain untuk fungsi Authentication, Authorization, dan Accounting (AAA).

2.2.2. Proxy Call Session Control Function (P-CSCF)

P-CSCF adalah point pertama dimana terjadi kontak antara user equipment dengan IMS. P-CSCF adalah SIP proxy yang meneruskan SIP messages antara user equipment dengan node yang benar pada IMS. P-CSCF juga berperan dalam otorisasi, QoS manajemen, dan mengamankan komunikasi antara dirinya dengan user equipment. Untuk melakukan hal ini P-CSCF akan berhubungan dengan user equipment pada G_m reference point menggunakan protokol SIP.

2.2.3. Interrogating Call Session Control Function (I-CSCF)

Peran utama dari I-CSCF adalah untuk mengalokasikan S-CSCF yang tepat untuk penggunaannya. I-CSCF membutuhkan informasi mengenai lokasi, kapabilitas, dan availabilitas dari masing-masing S-CSCF untuk membantu I-

CSCF tersebut dalam menentukan S-CSCF untuk pelanggan IMS. Setelah S-CSCF yang sesuai telah terpilih, I-CSCF akan menginformasikan S-CSCF yang harus dipilih oleh pengguna berdasarkan data di HSS.

2.2.4. Serving Call Session Control Function (S-CSCF)

S-CSCF berfungsi untuk menyediakan layanan kepada pengguna. Pada implementasinya S-CSCF berperan seperti registrasi SIP. S-CSCF akan menerima atau menolak permintaan dari user equipment. Jika sudah terautentikasi, pengguna akan diizinkan untuk menggunakan network resources sesuai dengan profilnya masing-masing. S-CSCF juga dapat berperan sebagai SIP proxy biasa yang meneruskan SIP request atau SIP reply ke endpoint yang dituju. S-CSCF terhubung dengan P-CSCF pada M_w reference point menggunakan protokol SIP.

2.2. VIDEO ON DEMAND

Video on Demand adalah suatu sistem yang memungkinkan untuk memilih dan menonton/mendengar video sesuai dengan permintaan pengguna. *Video on Demand* mengkombinasikan kualitas layanan dari TV kabel dan kemampuan interaksi dari VCR. Dengan layanan VoD, para pengguna dapat memilih video yang akan mereka tonton, bisa menonton kapan saja mereka mau, dan dapat menikmati fasilitas interaksi layaknya VCR/DVD seperti *fast forward* dan *rewind*. Inilah yang membedakan VoD dengan IPTV, pada IPTV pengguna tidak dapat menentukan video apa yang akan mereka tonton, mereka hanya bisa memilih *channel* apa yang sedang diputar oleh para penyedia layanan. Selain itu para pengguna juga tidak dapat menentukan kapan video itu dimulai dan tidak dapat menikmati fasilitas interaksi seperti VCR.

Jadi dapat disimpulkan bahwa VoD memiliki dua fitur penting, yaitu:

- Pengguna layanan dapat memilih video yang ingin mereka tonton dari *digital library*, menentukan waktu dan posisi dimana mereka akan memulai video tersebut.
- Fungsi-fungsi interaksi dari VCR/DVD seperti *start*, *pause*, *stop*, *forward*, *backward*, *fast forward*, dan *fast rewind* dapat disediakan

oleh layanan VoD karena pada layanan ini digunakan protokol RTSP.

Pada pengimplementasiannya VoD dapat diklasifikasikan ke dalam dua kelompok, tergantung dari apakah konten dari VoD itu distreamingkan atau didownload:

- *Streaming VoD*, adalah layanan VoD yang *real time*, video diidownload langsung dari server ke user equipment, kebutuhan QoS transport sama dengan IPTV. Bandwidth dan kapasitas processing yang besar sangat dibutuhkan pada layanan ini.
- *Downloading VoD*, adalah layanan *best effort*, video didownload dan disimpan terlebih dahulu baru ditampilkan pada *user equipment*. *Hard disk* dimana video disimpan terlebih dahulu dapat berada pada sisi penyedia layanan ataupun pada sisi pengguna.

2.3. PROTOCOL-PROTOCOL PADA LAYANAN MULTIMEDIA

2.3.1. Real Time Transport Protocol (RTP)

RTP adalah suatu standard untuk mengirimkan data multimedia secara *real-time* seperti audio dan video. RTP menyediakan layanan penyampaian *end to end* untuk data yang mempunyai karakteristik yang *real-time*, seperti audio dan video interaktif. RTP terdiri dari suatu data dan *control part* yang disebut RTCP. RTP merupakan protokol pada *layer application* yang berjalan di atas UDP tapi bisa juga di atas protokol lain.

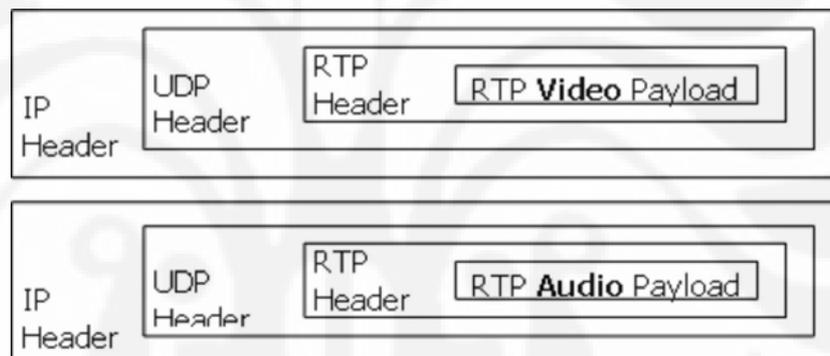
RTP menyediakan servis pengiriman data *end-to-end real-time*. Servis ini meliputi *payload type identification*, *sequence numbering*, dan *time stamping*. Jaringan tempat pengiriman paket kemungkinan besar akan menyebabkan *delay* dengan variasi waktu tertentu. RTP didesain untuk membawa informasi waktu bersama dengan data. Dengan adanya *timestamp* maka memungkinkan untuk *re-time* data yang diterima dengan *source timing* dan dengan akurasi yang cukup baik bagi sebagian besar aplikasi multimedia *streaming*.

Disamping *delay* yang tadi telah disebutkan, pada IP network juga memungkinkan terjadinya kehilangan paket atau *packet loss*. Untuk itu *sequence*

number dimasukkan ke dalam RTP. Dengan adanya *sequence number* maka penerima paket dapat mengetahui berapa paket yang hilang pada saat pengiriman.

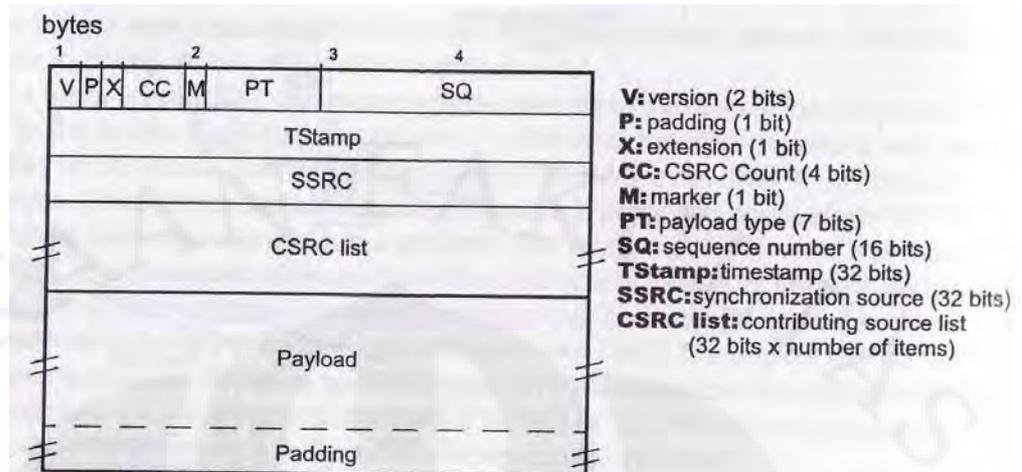
Fungsi penting lainnya dari RTP adalah *payload identification*. Pada implementasinya data dengan *payload* yang berbeda akan diperlakukan berbeda pada penerima sesuai dengan tipe paket. Untuk itu RTP akan memberitahu penerima tentang tipe data yang dikirim pada *payload* tersebut dan bagaimana data tersebut diencoded.

Informasi RTP dienkapsulasi dalam packet UDP. Jika paket RTP hilang atau didrop di jaringan, maka RTP tidak akan melakukan *retransmission* (sesuai standard protocol UDP). Hal ini agar *user* tidak terlalu lama menunggu (*long pause*) atau *delay*, dikarenakan permintaan *retransmission*.



Gambar 2.2 Enkapsulasi RTP

Sebuah pesan RTP mengandung RTP *header* yang diikuti dengan RTP *payload*. Gambar di bawah ini menunjukkan *header* RTP yang memiliki struktur sebagai berikut [2]



Gambar 2.3 RTP Packet Format

Sumber: Triple Play Building The Converges Network For IP, VoIP, and IPTV

- *Version (V)*: Field ini merupakan versi dari RTP.
- *Padding (P)*: Field ini digunakan jika media *stream* di-enkripsi.
- *Extension (X)*: Field ini merupakan ekstensi tambahan yang mengikuti *header* yang dibuat oleh tipe *payload* tertentu.
- *CSCR count (CC)*: Field ini memuat nomor CSRC.
- *Marker (M)*: berfungsi menandai awal dari *frame* baru pada video.
- *Payload Type (PT)*: merupakan 7-bit field yang menandai *codec* yang digunakan.
- *Sequence Number*: berukuran 16 bit yang berfungsi untuk mendeteksi hilangnya paket karena jumlah *sequence number* akan bertambah untuk setiap paket RTP yang dikirim.
- *Timestamps*: berupa 32 bit yang mengindikasi waktu relatif ketika *payload* disample.
- *Synchronization Source Identifier (SSRCI)*: mengidentifikasi *sender* dari paket RTP.
- *CSCR Contributing Source Identifier*: field ini hanya ada jika paket RTP telah dikirim oleh *mixer*.

2.3.2. Real Time Control Protocol (RTCP)

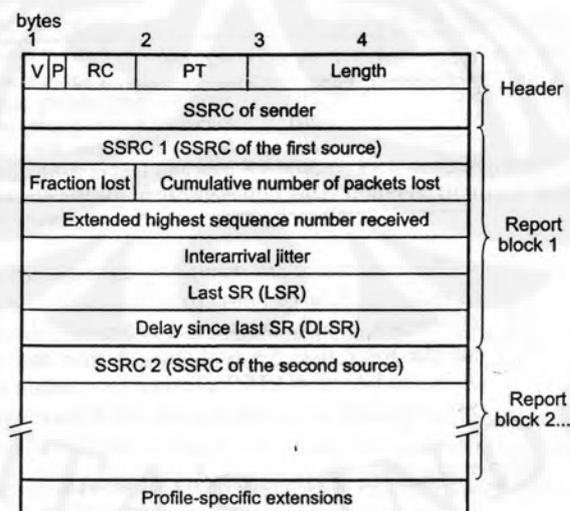
RTCP merupakan protokol komplemen dari RTP yang mempunyai fungsi utama untuk memberikan *feedback* tentang kualitas transmisi. Pada implementasinya RTCP dimultiplex bersama dengan RTP dimana RTP menggunakan nomor port genap dan RTCP menggunakan nomor port ganjil pada saat dikirimkan di dalam UDP [3]. Protokol ini memungkinkan partisipan pada suatu sesi RTP saling mengirimkan laporan kualitas. Fungsi laporan tersebut adalah mengetahui kualitas dari koneksi yang dibuat termasuk informasi seperti jumlah paket yang dikirim dan diterima, jumlah paket yang hilang dan jitter dari paket. RTCP paket memiliki 5 tipe sebagai berikut:

Type	PT id	Purpose
SR	200	Sender report: quality statistics from active senders
RR	201	Receiver report: quality statistics from receivers that are not senders
SDES	202	Source description items: information on the transmitter
BYE	203	Indicates end of participation
APP	204	Application-specific functions

Gambar 2.4 RTCP Packet Types

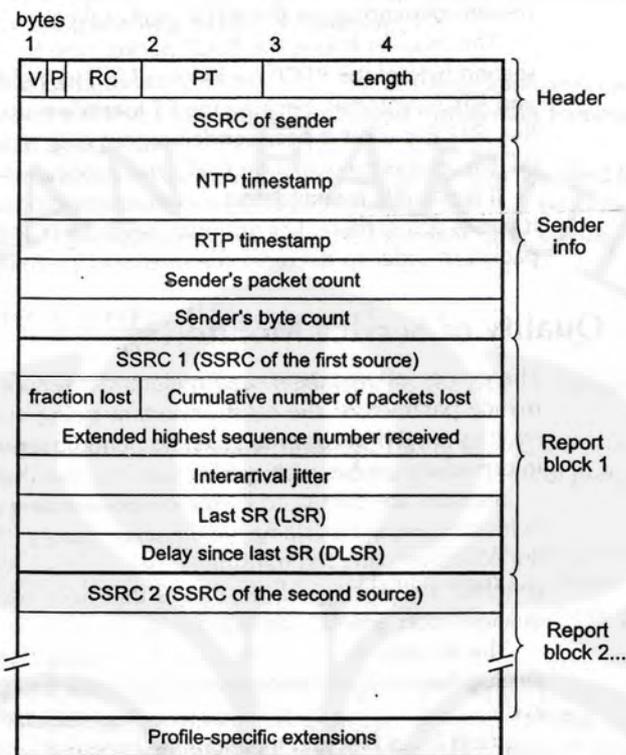
Sumber: Triple Play Building The Converges Network For IP, VoIP, and IPTV

Sesuai yang telah disebutkan tadi, fungsi utama dari RTCP adalah sebagai QoS *monitoring*. RTCP paket yang paling berperan untuk melakukan hal ini adalah Sender Report (SR) dan Receiver Report (RR).



Gambar 2.5 RTCP RR Packet Format

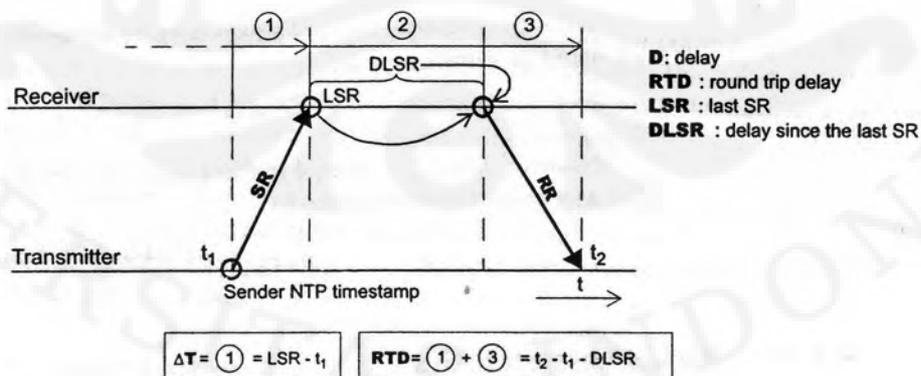
Sumber: Triple Play Building The Converges Network For IP, VoIP, and IPTV



Gambar 2.6 RTCP SR Packet Format

Sumber: Triple Play Building The Converges Network For IP, VoIP, and IPTV

Untuk memonitoring packet loss RTP menyimpan informasi mengenai jumlah paket pada *sender's packet count filed* di SR dan jumlah byte dari payload yang dikirim pada *sender's byte count filed* di SR. Sehingga nantinya RTCP dapat mengkalkulasikan jumlah paket yang hilang pada *cumulative number of packet lost field*.



Gambar 2.7 Perhitungan One Way Delay dan Round Trip Delay

Sumber: Triple Play Building The Converges Network For IP, VoIP, and IPTV

One Way Delay dapat dihitung dengan melihat perbedaan waktu dari *originating packet timestamp* dengan waktu tibanya paket SR. Nantinya penerima juga akan mengirimkan RR ke pengirim sehingga memungkinkan untuk pengkalkulasian *Round Trip Delay*.

2.3.3. Real Time Streaming Protocol (RTSP)

RTSP (*Real Time Streaming Protocol*) adalah protokol pada *layer application* digunakan oleh program *streaming* multimedia untuk mengatur pengiriman data secara *real-time*. Keuntungan RTSP adalah bahwa protokol ini menyediakan koneksi yang memiliki status antara *server* dan klien, yang dapat mempermudah klien ketika ingin melakukan *pause* atau mencari posisi *random* dalam *stream* ketika memutar kembali data. Biasanya diterapkan pada pengiriman *video on demand*.

RTSP memiliki empat buah perintah. Perintah ini dikirim dari client ke sebuah *server streaming* RTSP. Keempat perintah tersebut adalah sebagai berikut[4]:

- **Setup**, yaitu *server* mengalokasikan sumber daya kepada sesi klien.
- **Play**, yaitu *server* mengirim sebuah *stream* ke sesi klien yang telah dibangun dari perintah *setup* sebelumnya.
- **Pause**, yaitu *server* menunda pengiriman stream namun tetap menjaga sumber daya yang telah dialokasikan.
- **Teardown**, yaitu *server* memutuskan koneksi dan membebaskan tugas sumber daya yang sebelumnya telah digunakan.

2.3.4. Session Initiation Protocol (SIP)

SIP adalah *peer-to-peer* signaling protokol, dikembangkan oleh *Internet Engineering Task force* (IETF), yang mengizinkan endpoint-nya untuk memulai dan mengakhiri sesi komunikasi [5]. Di dalam IP dan *telephone* tradisional, selalu dibedakan dengan jelas dua tahap panggilan *voice*. Tahap pertama adalah “*Call Setup*” yang mencakup semua detail keperluan agar dua perangkat *telephone* dapat berkomunikasi. Tahap selanjutnya adalah “*transfer data*” dimana *call setup* sudah terbentuk. SIP sangat fleksibel dan didesain secara general untuk *setup*

real-time multimedia sessions antara *group participants*. Sebagai contoh, selain untuk *call telephone* yang sederhana, SIP dapat juga digunakan untuk *set-up conference video* dan audio atau *instant messaging*.

Arsitektur dari SIP terdiri dari dua komponen yaitu *user agent* dan *servers*. *User agent* merupakan *end point* dari sistem dan memuat dua sub sistem yaitu *user agent client* (UAC) yang membangkitkan *request*, dan *user agent server* (UAS) yang merespon *request*. *SIP server* adalah kesatuan fungsi *logic*, dimana tidak perlu memisahkan alat secara fisik. Fungsi dari empat *server* tersebut yaitu:

- *Proxy Server* : merupakan host jaringan yang berperan sebagai perantara yang bertujuan untuk meminta *request* atas nama *client* yang lain. *Proxy* harus bertindak sebagai *server* dan *client*, dia harus mengarahkan SIP request pada *user agent server*, dan mengarahkan SIP respons pada *user agent client* . *Proxy server* juga berfungsi untuk melakukan *routing*, memastikan *request* disampaikan pada yang berhak menerima, dan juga membuat kebijakan seperti menyakinkan bahwa pemakai tertentu diijinkan untuk melakukan panggilan.
- *Redirect Server*: merupakan kesatuan logika yang mengarahkan satu klien pada perangkat pengganti dari *Uniform Resource indicators* (URIs) untuk menyelesaikan tugas *request*.
- *Registrar Server* : menerima dan memproses pesan pendaftaran yang mengijinkan lokasi dari suatu *endpoint* dapat diketahui keberadaannya. *Registrar Server* ini kerjanya berhubungan dengan *Location Server*.
- *Location Server* : menyediakan *service* untuk *database* abstrak yang berfungsi mentranslasikan alamat dengan kata / keterangan yang ada pada domain jaringan.

Messages yang terdapat pada SIP didefinisikan dalam dua format :

- *Request*, dikirim dari *client* ke *server* , yang berisi tentang operasi yang diminta oleh *client* tersebut.
- *Responses* , dikirim dari *server* ke *client*, yang berisi informasi mengenai status dari apa yang diminta oleh *client*.

Ada enam tipe dari *request messages* :

- INVITE : menunjukkan bahwa *user* atau *service* sedang diundang untuk bergabung dalam *session*. Isi dari pesan ini akan memasukkan suatu uraian menyangkut *session* untuk *user* yang akan diundang.
- ACK : mengkonfirmasi bahwa *client* telah menerima suatu *final response* untuk suatu INVITE *request* , dan hanya digunakan di INVITE *request*.
- OPTION : digunakan untuk *query* suatu *server* tentang kemampuan yang dimilikinya.
- BYE : dikirim oleh *user agent client* untuk menunjukkan pada *server* bahwa percakapan ingin segera diakhiri.
- CANCEL : digunakan untuk membatalkan suatu *request* yang sedang menunggu keputusan.
- REGISTER : digunakan oleh *client* untuk mendaftarkan informasi kontak.

Response messages berisi status kode dan keterangan tentang kondisi dari *request* tersebut. Nilai-nilai dari kode status yang serupa dengan penggunaan pada HTTP, dibagi dalam enam kategori:

- **1xx**: *Provisional*, *request* telah diterima dan sedang melanjutkan proses.
- **2xx**: *Success*, tindakan dengan sukses diterima, dipahami, dan disetujui.
- **3xx**: *Redirection*, tindakan lebih lanjut diperlukan untuk memproses permintaan ini.
- **4xx**: *Client Error*, permintaan berisi sintak yang salah dan tidak bisa dikenali oleh *server* sehingga *server* tidak dapat memprosesnya.
- **5xx**: *Server Error*, *server* gagal untuk memproses suatu permintaan yang sah.
- **6xx**: *Global Failure*, permintaan tidak dapat dipenuhi oleh *server* manapun.

2.3.5. Session Description Protocol (SDP)

Dalam melakukan *session establishment* SIP tidak mendeskripsikan sesi multimedia yang akan dibangun, fungsi tersebut dijalankan oleh protokol tersendiri yang disebut dengan *session description protocol* (SDP). SDP adalah

Universitas Indonesia

protokol layer aplikasi yang berfungsi untuk memberikan deskripsi tentang suatu sesi multimedia[6], deskripsi tersebut antara lain mengenai :

1. Nama dan tujuan *session*
2. Waktu aktif *session* tersebut
3. Tipe media yang akan digunakan
4. *Address, port*, dll

Secara umum, protokol SDP digunakan pada saat melakukan *session announcement* serta *session invitation*. Informasi yang diberikan oleh sebuah pesan SDP antara lain adalah nama *session* dan tujuan penggunaan *session*, waktu aktif dari sebuah *session*, jenis media yang digunakan, format media, dan informasi untuk menerima media tersebut. Berikut adalah isi dari pesan SDP, *field-field* yang bersifat *optional* diberi tanda *[7].

Session Description

v= (protocol version)
 o= (owner/creator and *session* identifier)
 s= (*session* name)
 i=* (*session* information)
 u=* (URI of description)
 e=* (email address)
 p=* (phone number)
 c=* (connection information)
 b=* (bandwidth information)

One or more time description

z=* (time zone adjustments)
 k=* (encryption key)
 a=* (zero or more *session* attributes lines)

Time description

t=* (time the *session* is active)
 r=* (zero or more repeat times)

Media description

m= (media name and transport address)
 i=* (media title)

Dalam penggunaannya SIP mencantumkan SDP pada bagian *message body*.

2.4. PARAMETER QUALITY OF SERVICE (QOS)

Pada jaringan *Video on Demand* ini ada 4 parameter Quality of Service atau QoS yang menentukan kualitas *Video on Demand* ini, yaitu[8]:

2.4.1. Delay

Delay adalah waktu yang dibutuhkan oleh sebuah paket data terhitung dari saat pengiriman oleh transmitter sampai saat diterima oleh receiver. Beberapa jenis delay diantaranya adalah sebagai berikut:

- *Propagation delay* (*delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima)
- *Serialization delay* (*delay* pada saat proses peletakan bit ke dalam *circuit*)
- *Processing delay* (*delay* yang terjadi saat proses *coding*, *compression*, *decompression* dan *decoding*)
- *Packetization delay* (*delay* yang terjadi saat proses paketisasi *digital voice sample*)
- *Queuing delay* (*delay* akibat waktu tunggu paket sampai dilayani)
- *Jitter buffer* (*delay* akibat adanya *buffer* untuk mengatasi *jitter*)

2.4.2. Jitter

Jitter merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. Untuk mengatasi *jitter* maka paket data yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar.

2.4.3. Packet Loss

Packet loss adalah banyaknya paket yang hilang selama proses transmisi ke tujuan. Packet loss terjadi ketika *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu, maka *frame* (gabungan data payload dan header yang di transmisikan) data akan dibuang sebagaimana perlakuan terhadap *frame* data pada jaringan berbasis IP.

2.4.4. Troughput

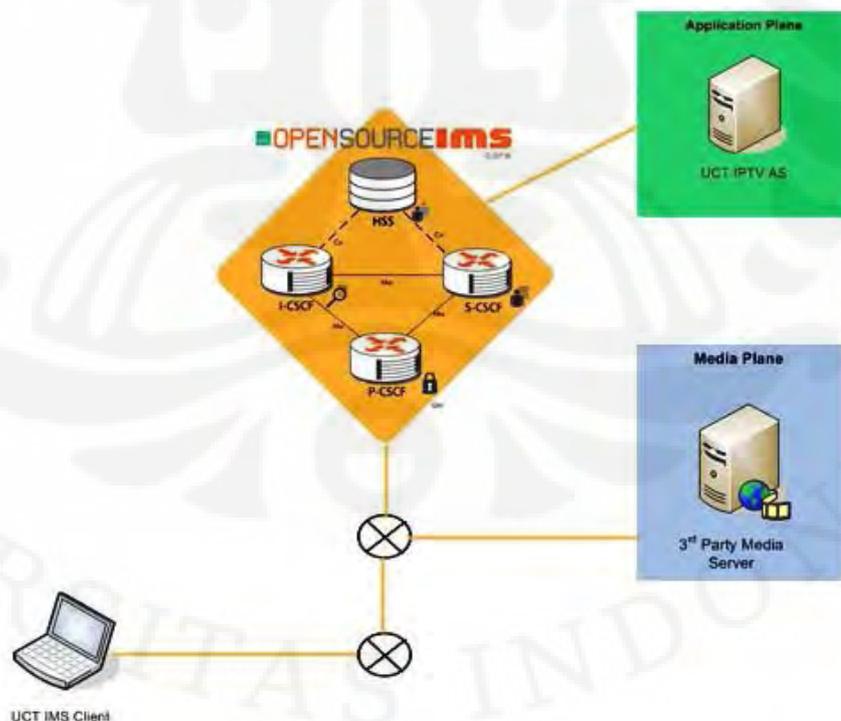
Troughput adalah jumlah bit atau paket dari suatu unit data yang diterima dengan benar oleh receiver.



BAB 3

PERANCANGAN VIDEO ON DEMAND BERBASIS OPEN IMS

Jaringan VoD yang dibuat dan diemulasikan pada penelitian ini adalah jaringan VoD yang berbasis IMS dalam hal ini OpenIMS. Pada jaringan ini terdapat 4 komponen utama, yaitu IMS Core, IMS Client, Application Server, dan Media Server. Dimana fungsi IMS Core pada jaringan VoD ini adalah memproses semua permintaan dari client dan meneruskannya ke application server. Application server nantinya akan melihat dan mencocokkan permintaan dari client dengan hash table yang ada padanya dan mencari alamat RTSP yang sesuai dengan permintaan client dan mengirimkannya ke client. Client yang telah mengetahui alamat RTSP yang ditujunya akan menginisiasi RTSP session dengan media server yang dituju. Secara garis besar proses ini dapat digambarkan sebagai berikut:



Gambar 3.1 Arsitektur VoD

Sumber: http://uctimsclient.berlios.de/uctiptv_advanced_howto.html

Jadi pada perancangan VoD berbasis Open IMS ini secara garis besar alurnya adalah sebagai berikut:

1. Client akan melakukan registrasi kepada IMS Core
2. IMS Core akan memeriksa apakah client tersebut sudah terdapat pada HSS atau tidak, jika sudah maka proses registrasi berhasil dan client diperbolehkan melakukan request selanjutnya.
3. Selanjutnya client akan melakukan SIP request untuk salah satu layanan VoD yang sudah tersedia
4. IMS Core akan menerima request dari client dan meneruskannya ke application server.
5. Application server akan menerjemahkan SIP request dari client menjadi alamat RTSP dari media server yang cocok dengan request dari client.
6. IMS core akan meneruskan jawaban dari application server ke client.
7. Client yang sudah tahu alamat RTSP dari layanan yang dia minta akan membangun sesi dengan media server secara langsung.

Untuk melakukan proses di atas maka dalam perancangan ini diperlukan konfigurasi dari beberapa komponen penting, yaitu IMS core, application server, media server, client, dan registrasi user-user baru.

3.1. Konfigurasi dan Instalasi Core IMS

Seperti yang telah dijelaskan sebelumnya, pada penelitian ini IMS core berfungsi untuk menerima request dari client, memeriksa apakah client tersebut sudah terdaftar dan mempunyai hak untuk mengakses layanan yang dia minta atau tidak, meneruskan request dari client ke application server, dan selanjutnya mengembalikan jawaban dari application server ke client.

Untuk pembuatan VoD berbasis IMS ini, core yang digunakan adalah OpenIMSCore yang sifatnya open source. Langkah-langkah instalasinya adalah sebagai berikut:

1. Update repositori ubuntu

Update repositori ubuntu sangat penting supaya kita dapat mendownload package-package terbaru yang telah dikembangkan

sampai saat ini. Cara mengupdatenya adalah dengan mengetikkan perintah berikut di terminal:

```
# sudo apt-get update
```

2. Download dan install Subversion package

Subversion adalah salah satu package yang diperlukan dalam penginstalan OpenIMSCore ini, cara mendapatkan package ini adalah dengan mengetikkan perintah berikut di terminal:

```
# sudo apt-get install subversion
```

3. Membuat folder OpenIMSCore

Folder OpenIMSCore ini nantinya digunakan sebagai tempat semua source code yang nantinya akan didownload. Selain itu status kepemilikan folder ini harus dirubah agar dapat dirubah-ubah sesuai kehendak user.

```
# sudo mkdir /opt/OpenIMSCore/
```

```
# sudo chown -R username /opt/OpenIMSCore/
```

4. Membuat folder ser_ims dan FHoSS di dalam folder OpenIMSCore

Folder ser-ims nantinya akan menjadi tempat source code CSCF dan FhoSS untuk tempat source code HSS

```
# cd /opt/OpenIMSCore
```

```
# mkdir ser_ims
```

```
# mkdir FhoSS
```

5. Download source code CSCF dan FHOSS

Untuk CSCF:

```
# svn checkout
```

```
http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk  
ser_ims
```

Untuk HSS:

```
# svn checkout
```

```
http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk
```

```
FHoSS
```

6. Install package-package yang dibutuhkan

Instalasi OpenIMSCoe ini membutuhkan beberapa package-package diantaranya sun-java-jdk, mysql-server, libmysqlclient15-dev, libxml2-dev, bind, ant, flex, dan bison.

```
# sudo apt-get install sun-java6-jdk mysql-server libmysqlclient15-dev libxml2 libxml2-dev bind9 ant flex bison
```

7. Copy open-ims DNS file ke folder bind

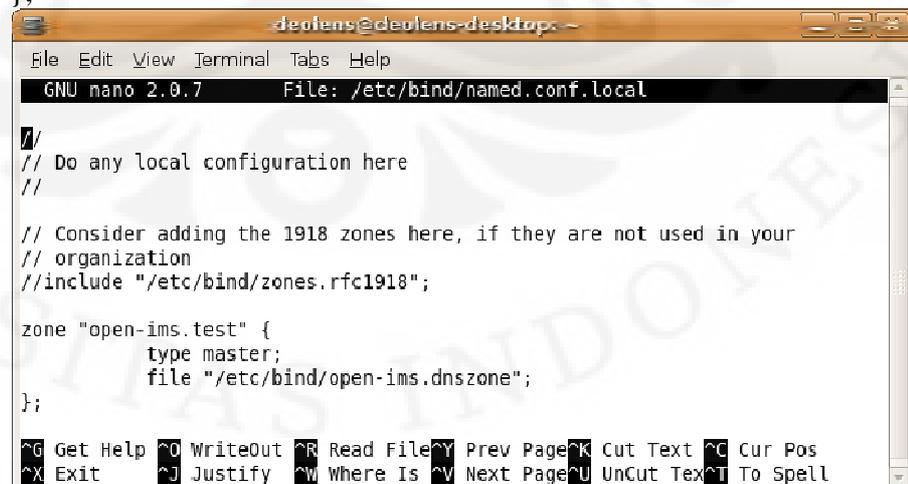
```
# sudo cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone /etc/bind/
```

8. Ubah file named.conf.local

```
# sudo nano /etc/bind/named.conf.local
```

Lalu tambahkan kata-kata berikut ke dalam file tersebut:

```
zone "open-ims.test" {
    type master;
    file "/etc/bind/open-ims.dnszone";
};
```



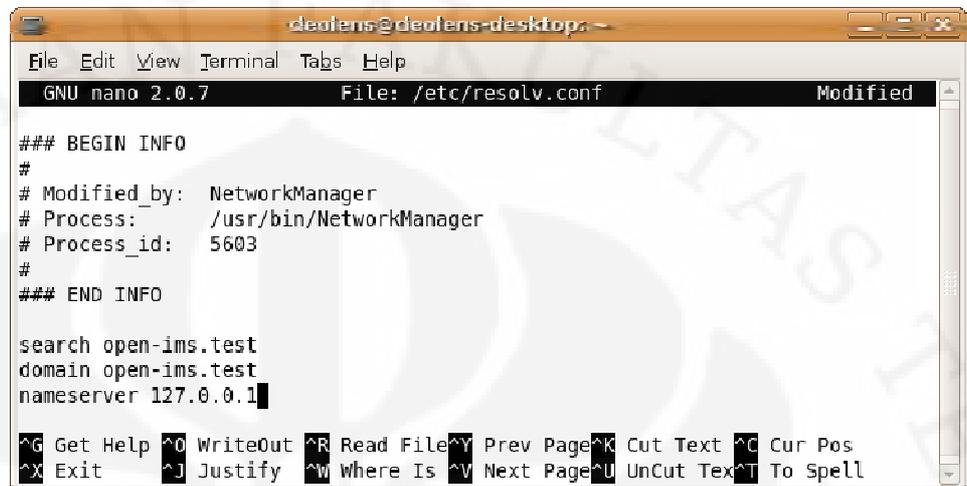
```
deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "open-ims.test" {
    type master;
    file "/etc/bind/open-ims.dnszone";
};
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Gambar 3.2 Tampilan File named.conf.local localhost

9. Ubah file resolv.conf

```
# sudo nano /etc/resolv.conf
```

Lalu ubah file tersebut menjadi seperti ini:

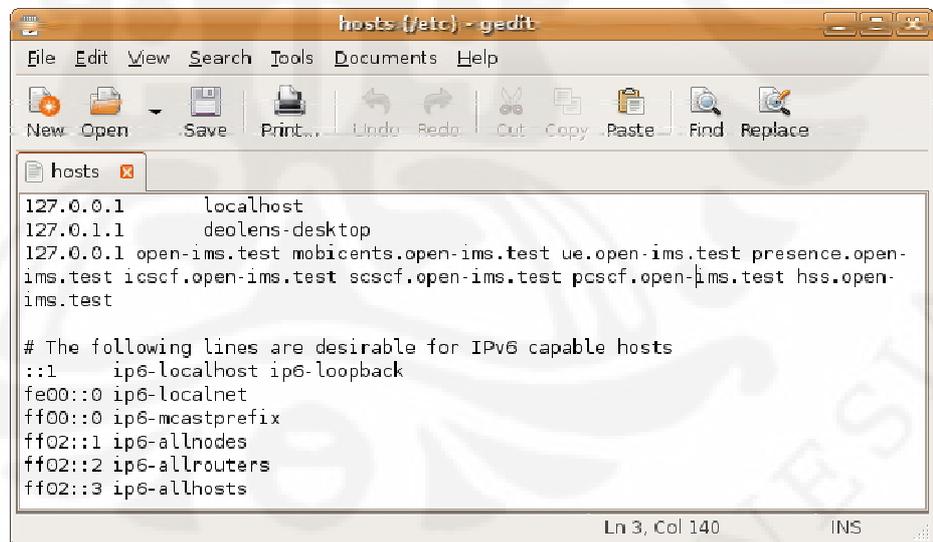


```
deolens@deolens-desktop
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/resolv.conf Modified
### BEGIN INFO
#
# Modified_by: NetworkManager
# Process: /usr/bin/NetworkManager
# Process_id: 5603
#
### END INFO
search open-ims.test
domain open-ims.test
nameserver 127.0.0.1
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Gambar 3.3 Tampilan File resolv.conf localhost

10. Ubah file hosts

```
# sudo nano /etc/hosts
```



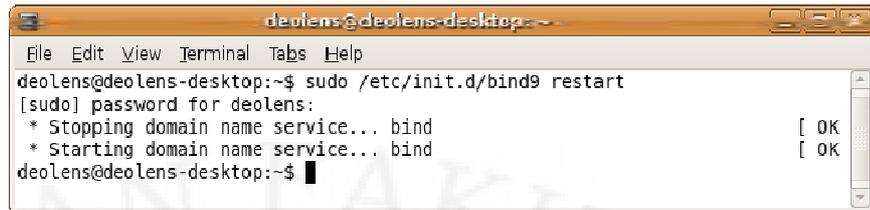
```
hosts (/etc) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
hosts
127.0.0.1 localhost
127.0.1.1 deolens-desktop
127.0.0.1 open-ims.test mobicents.open-ims.test ue.open-ims.test presence.open-ims.test
icscf.open-ims.test sscsf.open-ims.test pcscf.open-ims.test hss.open-ims.test
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
Ln 3, Col 140 INS
```

Gambar 3.4 Tampilan File hosts localhost

11. Restart bind server

```
# sudo /etc/init.d/bind9 restart
```

Jika langkah-langkah di atas sudah benar maka pada saat restart bind akan mendapat response OK seperti gambar berikut ini:



```

deolens@deolens-desktop:~$ sudo /etc/init.d/bind9 restart
[sudo] password for deolens:
* Stopping domain name service... bind
* Starting domain name service... bind
deolens@deolens-desktop:~$

```

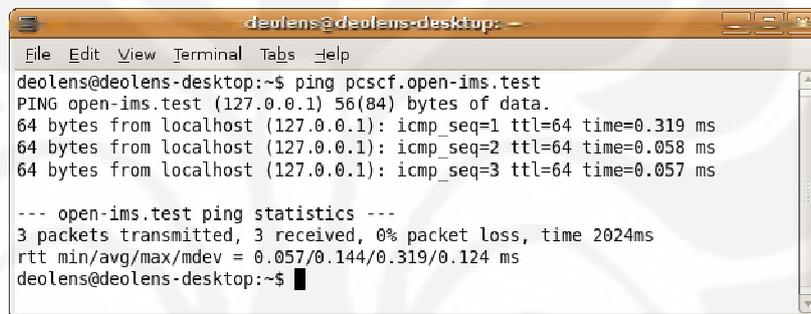
Gambar 3.5 Restart Bind

12. Cek apakah step 7 sampai 11 sudah benar

Ada 2 cara untuk memeriksanya yaitu:

ping pcscf.open-ims.test

Lihat apakah ping mendapat response, atau tidak



```

deolens@deolens-desktop:~$ ping pcscf.open-ims.test
PING open-ims.test (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.319 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.057 ms

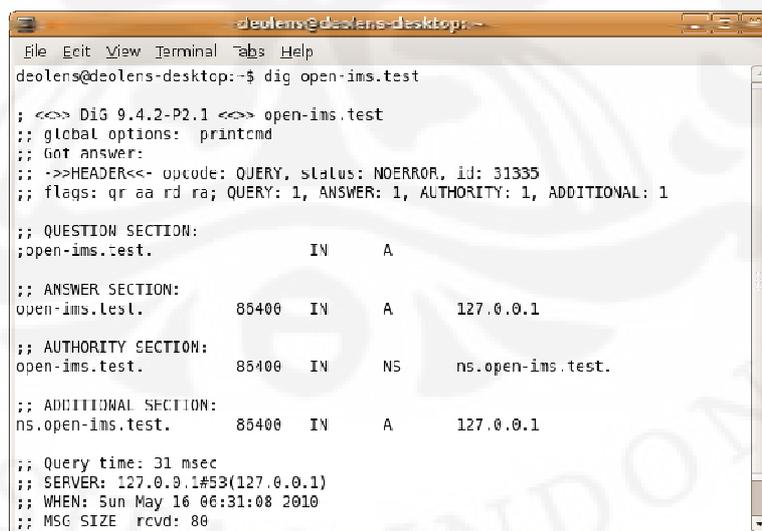
--- open-ims.test ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2024ms
rtt min/avg/max/mdev = 0.057/0.144/0.319/0.124 ms
deolens@deolens-desktop:~$

```

Gambar 3.6 Ping PCSCF

dig open-ims.test

Pada answer section harus tertulis 127.0.0.1



```

deolens@deolens-desktop:~$ dig open-ims.test

;<<< DiG 9.4.2-P2.1 <<< open-ims.test
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31335
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;open-ims.test.                IN      A

;; ANSWER SECTION:
open-ims.test.                85400   IN      A      127.0.0.1

;; AUTHORITY SECTION:
open-ims.test.                85400   IN      NS     ns.open-ims.test.

;; ADDITIONAL SECTION:
ns.open-ims.test.            85400   IN      A      127.0.0.1

;; Query time: 31 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun May 16 06:31:08 2010
;; MSG SIZE rcvd: 80

```

Gambar 3.7 dig open-ims.test

13. Setup database

```
# mysql -uroot -p < /opt/OpenIMSCore/ser_ims/cfg/icscf.sql
# mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/hss_db.sql
# mysql -uroot -p < /opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

14. Compile source code

Compile CSCF:

```
# cd /opt/OpenIMSCore/ser_ims
# sudo make install-libs all
```

Compile HSS:

```
# cd /opt/OpenIMSCore/ser_ims
# ant compile deploy
```

15. Copy configuration file ke folder OpenIMSCore

```
# cp /opt/OpenIMSCore/ser_ims/cfg/* /opt/OpenIMSCore/
```

16. Jalankan OpenIMSCore

Jalankan setiap CSCF pada satu tab terminal:

```
# cd /opt/OpenIMSCore/
# ./pcscf.sh
# ./icscf.sh
# ./scscf.sh
```

Jalankan FhoSS:

```
# cd /opt/OpenIMSCore/FhoSS/deply/
# export JAVA_HOME=/usr/lib/jvm/java-6-sun
# ./startup.sh
```

17. Daftar element dan port numbertnya

Element	Port Number
P-CSCF	4060
I-CSCF	5060

S-CSCF	6060
Diameter	3868, 3869, 3870

18. Merubah OpenIMSCore localhost ke spesifik IP

Untuk merubah OpenIMSCore dari localhost ke spesifik IP maka kita harus merubah file resolv.conf, hosts, dan open-ims.dnszone, yaitu dengan mengganti semua IP localhost 127.0.0.1 menjadi spesifik IP yang kita gunakan. Pada skripsi ini digunakan IP 192.168.100.5

sudo nano /etc/resolv.conf



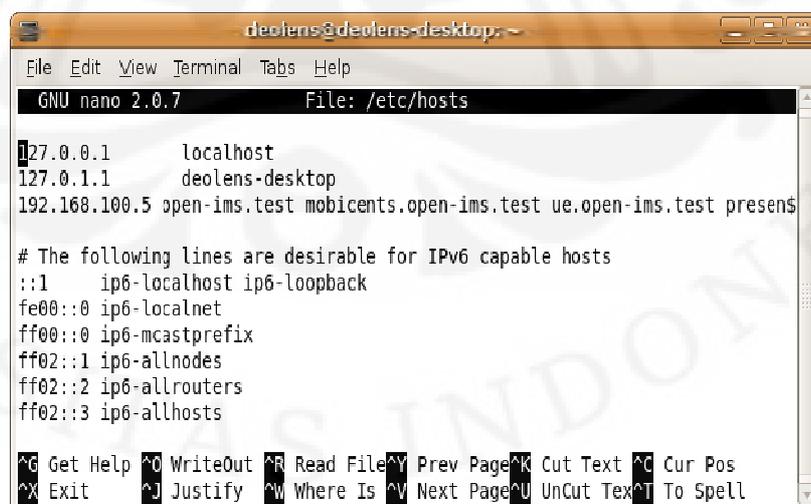
```

deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/resolv.conf
search open-ims.test
domain open-ims.test
nameserver 192.168.100.5
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.8 Tampilan File resolv.conf spesifik IP

sudo nano /etc/hosts



```

deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/hosts
127.0.0.1 localhost
127.0.1.1 deolens-desktop
192.168.100.5 open-ims.test mobicents.open-ims.test ue.open-ims.test presens
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.9 Tampilan File hosts spesifik IP

```
# sudo nano /etc/bind/open-ims.dnszone
```

```

deolens@deolens-desktop:
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/bind/open-ims.dnszone

$ORIGIN open-ims.test.
$TTL 1w
@           1D IN 50A    localhost. root.localhost. (
                2006101001 ; serial
                3H      ; refresh
                15M     ; retry
                1w      ; expiry
                1D )    ; minimum

ns          1D IN NS     ns
           1D IN A     192.168.100.5

pcscf       1D IN A     192.168.100.5
_sip.pcsf   1D SRV 0 0 4060 pcscf
_sip_udp.pcsf 1D SRV 0 0 4060 pcscf
_sip_tcp.pcsf 1D SRV 0 0 4060 pcscf

icscf       1D IN A     192.168.100.5
[ Read 61 lines ]
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.10 Tampilan File open-ims.dnszone spesifik IP

Setelah itu rubah settingan dari masing-masing node dengan command:

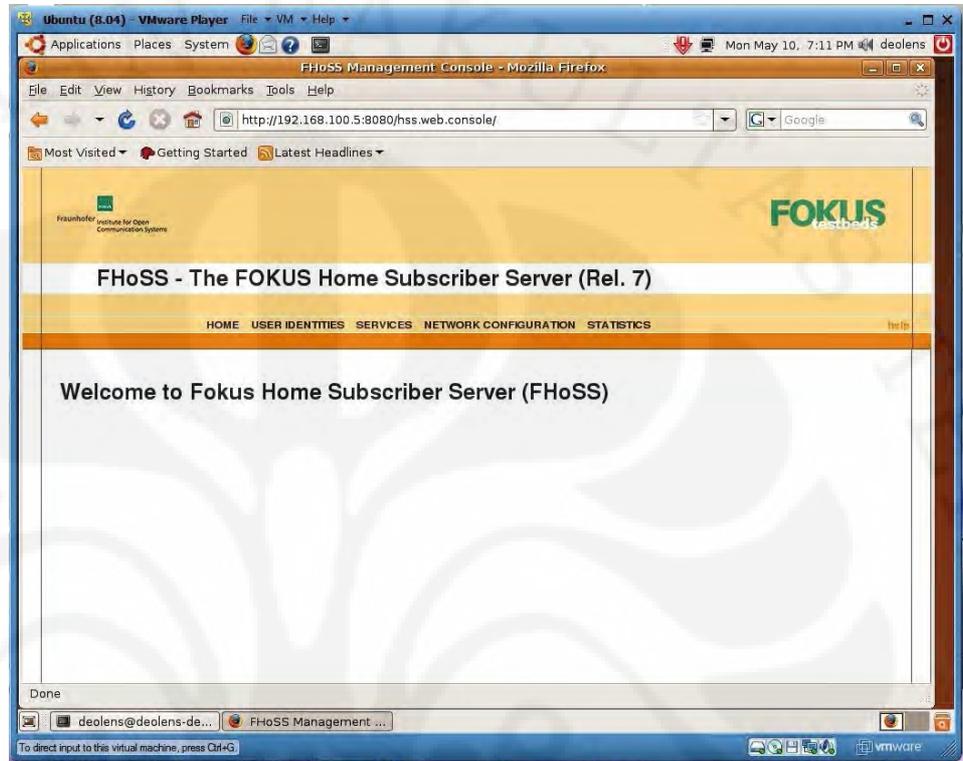
```
# cd /opt/OpenIMSCore/
# ./configurator.sh pcscf.cfg icscf.cfg icscf.xml scscf.cfg scscf.xml
ser_ims/cfg/icscf.sql FHoSS/deploy/DiameterPeerHSS.xml
FHoSS/deploy/hss.properties FHoSS/scripts/hss_db.sql
FHoSS/scripts/userdata.sql
```

Setelah menjalankan commnd tersebut maka user akan diminta memasukkan IP address dan realm. Pada penelitian ini IP address yang digunakan adalah 192.168.100.5 dan realmnya adalah open-ims.test

19. Ulangi langkah ke-11, ke-13, dan ke-16

20. Buka web browser dan ketikkan alamat IP dan port dari OpenIMSCore. Untuk penelitian ini digunakan IP address 192.168.100.5 dan port 8080 maka pada web browser ketikkan 192.168.100.5:8080.

21. Jika proses instalasi berhasil maka pada web browser akan muncul tampilan seperti berikut:



Gambar 3.11 Tampilan FHOSS

3.2. Instalasi dan Konfigurasi Application Server

Seperti yang telah dijelaskan di awal tadi, application server berfungsi untuk meroutingkan request dari client ke alamat RTSP dari media server yang dituju. Oleh karena itu application server yang digunakan pada VoD ini disebut indirection application server. Langkah pertama untuk menginstal application server ini adalah mendownload file installernya atau debian packagenya dari https://developer.berlios.de/project/showfiles.php?group_id=7844. Lalu install debian package tersebut.

Langkah selanjutnya adalah melakukan konfigurasi pada FhoSS seperti berikut ini:

1. Buat application server dengan nama iptv yang berjalan pada port 8010. Pada skripsi ini konfigurasi application server yang digunakan adalah sebagai berikut:

Permission for	UDR	PUR	SHR
Allowed Request	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Repository-Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MPU	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MS User State	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
S-CSCF Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IFC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Location	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
User-State	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Charging-Info	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MS-SDN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PSI Activation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DSA1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aliases Rep Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

ID	IFC Name	Detach
2	iptv_filter	Detach

Gambar 3.12 VoD Application Server

2. Buat trigger point dengan nama IPTV_trigger

Trigger Point -TP-

ID	IFC Name	Detach
2	iptv_filter	Detach

Add SPTs to Trigger Point

Not	<input type="checkbox"/>	SIP Method	INVITE	Delete
OR				
		Request-URI		
AND				
Not	<input type="checkbox"/>	SIP Header	To	Delete
		SIP Header Content	*media.open-ims.test.*	
OR				
Not	<input type="checkbox"/>	SIP Header	To	Delete
		SIP Header Content	*iptv.open-ims.test.*	
OR				
		Request-URI		
AND				
		Request-URI		

Gambar 3.13 VoD Trigger Point

Universitas Indonesia

- Hubungkan application server dan trigger point dengan initial filter criteria

Initial Filter Criteria -iFC-

ID	<input type="text"/>
Name*	iptv_filter
Trigger Point	IPTV_trigger
Application Server*	iptv
Profile Part Indicator	Any

Mandatory fields were marked with "*"

Save Refresh Delete

Gambar 3.14 iFC

- Masukkan iFC yang tadi dibuat (iptv_filter) ke Shared iFC Set

Shared iFC Sets -Sh-iFC-

ID-Set	<input type="text"/>
Name*	default_shared_set

Mandatory fields were marked with "*"

Save Refresh Delete

Attach iFC

Select iFC...	Priority	0	Attach
---------------	----------	---	--------

Warning: Priority values defined here can overwrite priority values defined in SP-iFC setup!

List of attached iFCs

ID	Name	Priority	Detach
1	default_ifc	0	Detach
2	iptv_filter	2	Detach

Gambar 3.15 Shared iFC Set

5. Masukkan iFC dan Shared iFC Set ke default Service Profile

Service Profile -SP-

ID	1
Name*	default_sp
Core Network Service Auth	0

Mandatory fields were marked with "**"

Attach IFC

Select IFC...	Priority 0	Attach
---------------	------------	--------

Attach Shared-IFC-Set

Select Shared-IFC...	Attach
----------------------	--------

List of attached IFCs

ID	IFC Name	Priority	Detach
1	default_ifc	0	Detach
2	iptv_filter	2	Detach

List of attached Shared-IFC-Sets

ID-Set	Name	Detach
1	default_shared_set	Detach

Gambar 3.16 Service Profile

6. Rubah application server key value file

Key value file ini merupakan file yang menghubungkan request dari client dengan alamat RTSP dari media server yang dituju. Jadi pada file ini kita tinggal memasukkan alamat RTSP dari media server ke dalam value dalam file ini. Caranya adalah sebagai berikut:

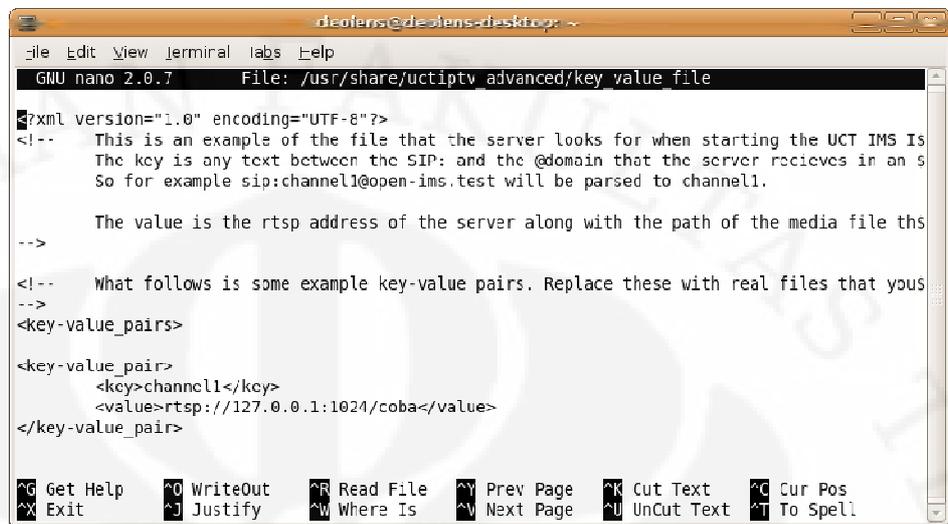
```
# sudo nano /usr/share/uctiptv_advanced/key_value_file
```

Maka akan keluar default key value file sebagai berikut:

```
<key-value_pairs>
  <key-value_pair>
    <key>channell</key>
    <value>rtsp://media_server_address.domain:8000
      /requested_channel</value>
  </key-value_pair>
</key-value_pairs>
```

Sekarang kita tinggal merubah isi di dalam <value></value> dengan alamat RTSP tempat media server berada. Pada skripsi ini

media server berada pada alamat RTSP 192.168.100.5:8000 sehingga key value file dirubah menjadi seperti berikut ini:



```

deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /usr/share/uctiptv_advanced/key_value_file
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is an example of the file that the server looks for when starting the UCT IMS Is
The key is any text between the SIP: and the @domain that the server recieves in an s
So for example sip:channel1@open-ims.test will be parsed to channel1.

The value is the rtsp address of the server along with the path of the media file ths
-->

<!-- What follows is some example key-value pairs. Replace these with real files that you
-->
<key-value_pairs>
<key-value_pair>
  <key>channel1</key>
  <value>rtsp://127.0.0.1:1024/coba</value>
</key-value_pair>

```

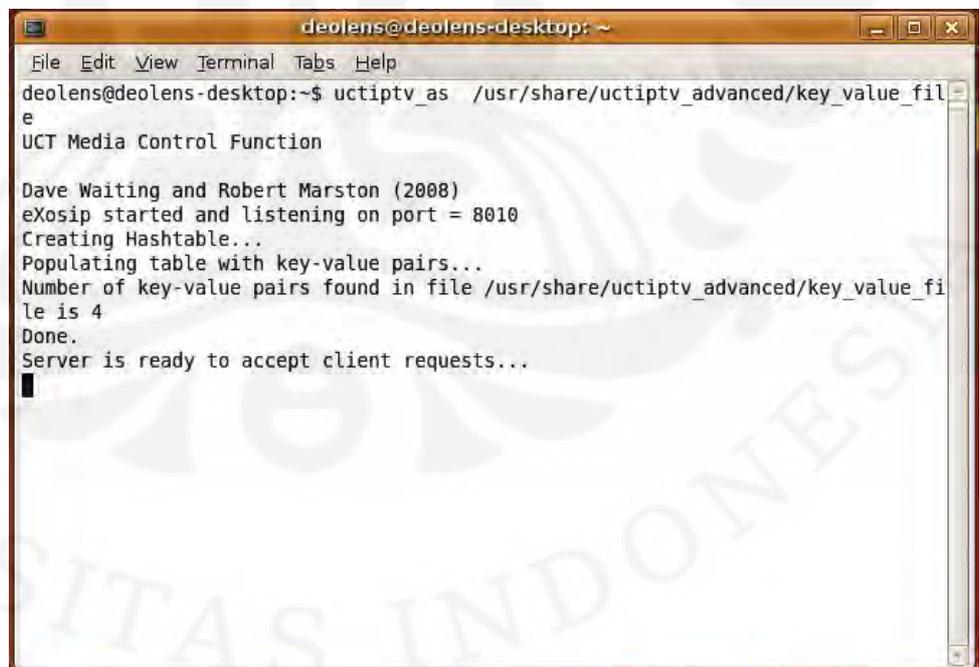
Gambar 3.17 Key Falue File

7. Jalankan Application Server

Ketikkan command berikut pada terminal:

```
# uctiptv_as /usr/share/uctiptv_advanced/key_value_file
```

Sehingga application server akan muncul seperti gambar berikut ini;



```

deolens@deolens-desktop: ~
File Edit View Terminal Tabs Help
deolens@deolens-desktop:~$ uctiptv_as /usr/share/uctiptv_advanced/key_value_file
UCT Media Control Function

Dave Waiting and Robert Marston (2008)
eXosip started and listening on port = 8010
Creating Hashtable...
Populating table with key-value pairs...
Number of key-value pairs found in file /usr/share/uctiptv_advanced/key_value_file is 4
Done.
Server is ready to accept client requests...

```

Gambar 3.18 Tampilan Application Server Pada Terminal

3.3. Instalasi dan Konfigurasi Media Server

Pada penelitian mengenai VoD berbasis IMS ini, media server berperan sebagai server untuk men-streaming-kan film-film atau video-video yang nantinya akan di-request oleh client. Pada penelitian ini digunakan VLC sebagai media server. Untuk men-download dan meng-install VLC beserta plugin-plugin-nya digunakan perintah berikut:

```
# sudo apt-get install vlc mozilla-plugin-vlc vlc-plugin-esd
```



Gambar 3.19 VLC Media Player

Langkah selanjutnya adalah membuat file konfigurasi dari film atau video yang akan di-streaming-kan. Pada penelitian ini file konfigurasi yang akan dibuat diberi nama vod.vlm, shortcut coba, dan nama video final.MPG yang bertempat di folder /home/deolens/Videos. File tersebut dibuat dengan cara mengetikkan perintah berikut pada terminal:

```
# sudo nano vod.vlm
```

Lalu masukkan kalimat-kalimat berikut ini pada file tersebut:

```
new coba vod enabled
```

```
setup coba input "/home/deolens/Videos/final.MPG"
```

Selanjutnya adalah men-streaming-kan file video yang telah kita siapkan ke dalam suatu alamat RTSP tertentu. Pada skripsi ini video yang telah disediakan akan di-streamingkan ke alamat RTSP 192.168.100.5 port 8000. Maka pada terminal ketikkan perintah berikut:

```
# vlc -vvv --ttl 12 --intf telnet --rtsp-host 192.168.1100.5:8000 --vlm-conf vod.vlm
```

3.4. Instalasi dan Konfigurasi IMS Client

Pada penelitian ini software yang digunakan sebagai client adalah UCT IMS Client. Untuk meng-install UCT IMS Client ini kita dapat men-download debian package-nya dari <http://uctimsclient.berlios.de/>. Setelah berhasil di-install maka untuk menjalankan UCT IMS Client ini kita tinggal mengetikkan perintah berikut pada terminal:

```
# sudo uctimsclient
```



Gambar 3.20 UCT IMS Client

Untuk me-request salah satu VoD kita tinggal mengetikkan alamat URI VoD tersebut. Contohnya pada penelitian ini akan di-request VoD dengan alamat Url sip:channel1@iptv.open-ims.test sehingga akan muncul tayangan video seperti berikut ini.



Gambar 3.21 Preview VoD

3.5. Pembuatan Subscriber-subscriber Baru

Pada Open IMS Core yang digunakan pada penelitian ini, di HSS hanya terdaftar dua default subscriber, yaitu Alice dan Bob. Oleh karena itu perlu ditambahkan subscriber-subscriber baru di luar kedua subscriber default tersebut. Penambahan subscriber baru ini dilakukan pada HSS melalui web browser pada bagian user identities. Secara umum penambahan subscriber baru ini dilakukan dalam 3 langkah, yaitu:

- Pembuatan IMS Subscription
- Pembuatan Private User Identity
- Pembuatan Public User Identity

Berikut ini adalah contoh pembuatan user baru:

- Membuat IMS Subscription dengan nama “deolens”

IMS Subscription -IMSU-

ID	<input type="text"/>	Create & Bind new IMPI
Name*	<input type="text" value="deolens"/>	
Capabilities Set	<input type="text" value="cap_set1"/>	
Preferred S-CSCF	<input type="text" value="scscf1"/>	
S-CSCF Name	<input type="text" value="sip.scscf.open-ims.test@0.0.0"/>	
Diameter Name	<input type="text" value="scscf.open-ims.test"/>	

Mandatory fields were marked with ""

Associate IMPI(s)

IMPI Identity

List of associated IMPIs

ID	IMPI Identity	Delete
5	deolens@open-ims.test	<input type="button" value="Delete"/>

Gambar 3.22 IMS Subscription

- Membuat Private User Identity dengan nama “deolens@open-ims.test”

Private User Identity -IMPI-

ID	5
Identity*	deolens@open-ims.test
Secret Key*	deolens
Authentication Schemes*	
Digest-AKAv1 (3GPP)	<input type="checkbox"/>
Digest-AKAv2 (3GPP)	<input type="checkbox"/>
Digest-MD5 (FOKUS)	<input type="checkbox"/>
Digest (CableLabs)	<input type="checkbox"/>
SIP Digest (3GPP)	<input type="checkbox"/>
HTTP Digest	<input type="checkbox"/>

Associate an IMSU

IMSU Identity

Associated IMSU

ID	IMSU Identity	Delete
3	deolens	<input type="button" value="Delete"/>

Create & Bind new IMPU

Associate IMPU(s)

IMPUS Identity

Warning: The current IMPI will be associated with all the corresponding IMPUS (within the same implicit-set)!

Gambar 2.23 Private User Identity

- Membuat Public User Identity dengan nama sip:deolens@open-ims.test”

Public User Identity -IMPU-

ID	3
Identity*	sip:deolens@open-ims.test
Barring	<input type="checkbox"/>
Service Profile*	default_sp
Implicit Set	3
Charging-Info Set	Select Charging-Info...
Can Register	<input checked="" type="checkbox"/>
IMPU Type*	Public_User_Identity
Wildcard PSI	
PSI Activation	<input type="checkbox"/>
Display Name	deolens manurung
User-Status	UN-REGISTERED

Mandatory fields were marked with "*"

Add Visited-Networks

Select Visited-Network...

List of Visited Networks

ID	Identity	Delete
1	open-ims.test	<input type="button" value="Delete"/>

Associate IMPI(s) to IMPU

IMPI Identity

Warning: This IMPI will be associated with all the corresponding IMPUS (within the same implicit-set)!

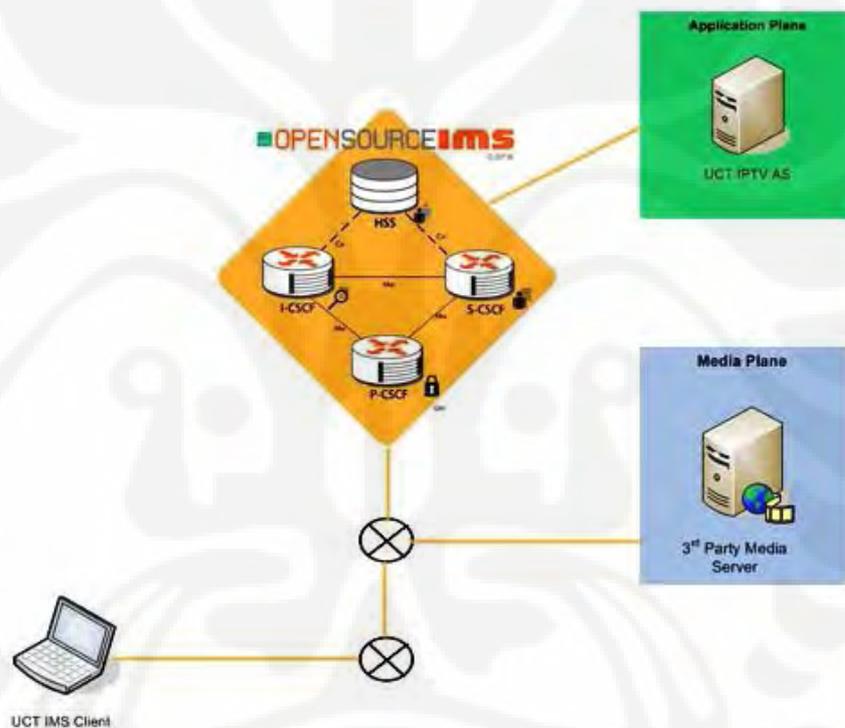
Gambar 2.24 Public User Identity

BAB 4

UJI COBA, PENGAMBILAN DATA, DAN ANALISIS

4.1. SKENARIO UJI COBA

Uji coba untuk layanan *Video on Demand* berbasis Open IMS ini akan dilakukan dalam 2 skenario utama sesuai dengan tujuan dari penelitian ini, yaitu untuk menganalisa *message flow* dari layanan *Video on Demand* ini dan untuk mengetahui pada *bandwidth* berapakah layanan *Video on Demand* ini dapat bekerja secara optimal jika mengacu pada standart ITU-T G 1010.



Gambar 4.1 Topologi Jaringan

Sesuai dengan Gambar 4.1, maka digunakan 2 buah *Personal Computer* (PC) sebagai router Vyatta, 1 buah Laptop sebagai *server* (OpenIMSCore), 1 buah laptop sebagai *client*:

Pada jaringan ini *client* diberi *IP Address* 192.168.1.x, *netmask* 255.255.255.0 dan *gateway* 192.168.1.1. Pada sisi router1 di mana terdapat 2 *interface* ethernet yaitu Eth0 yang terhubung ke router2 dan Eth1 yang terhubung

ke komputer *client*. Eth0 diberi *IP Address* 200.200.200.2, *netmask* 255.255.255.0. Eth1 yang diberi *IP Address* 192.168.1.1, *netmask* 255.255.255.0.

Pada sisi router2 terdapat 2 *interface* Ethernet juga yaitu Eth0 yang terhubung ke komputer *server* dan Eth1 terhubung ke router1. Ethernet Eth0 yang diberi *IP Address* 192.168.100.1, *netmask* 255.255.255.0 dan Eth1 yang diberi *IP Address* 200.200.200.1, *netmask* 255.255.255.0. Antara router1 dan router2 terhubung dengan menggunakan protokol *routing* RIP (*Routing Information Protocol*). Pada jaringan ini server diberi *IP Address* 192.168.100.5, *netmask* 255.255.255.0 dan *gateway* 192.168.100.1.

Skenario pertama dilakukan untuk mendapatkan data mengenai *message flow* pada saat registrasi *user*, pemanggilan salah satu layanan *Video on Demand*, pemutusan layanan *Video on Demand*, dan proses deregistrasi *user*. Pada uji coba ini *message flow* akan ditangkap dengan menggunakan salah satu *software network analyzer* yaitu Wireshark. Pada uji coba ini hanya digunakan 1 *client* dan pada jaringan tidak dilakukan pembatasan *bandwidth*.

Skenario kedua dilakukan untuk mengetahui pada *bandwidth* berapakah layanan *Video on Demand* ini dapat bekerja secara optimal. Untuk menganalisa hal ini maka *bandwidth* pada jaringan ini akan diatur dan dibatasi mulai dari 64kbps dan ditingkatkan sampai menemui *bandwidth* yang optimum. *Bandwidth* yang optimum disini akan didapatkan ketika parameter QoS layanan *Video on Demand* ini telah memenuhi standart ITU-T G 1010. Pada uji coba ini akan dilakukan dengan 1 *client*. Parameter pengujian yang akan diambil datanya adalah parameter QoS seperti *delay*, *jitter*, *packet loss*, dan *throughput*.

4.2. ANALISA MESSAGE FLOW

Pada uji coba unuk menganalisa *message flow* ini akan dibagi mejadi 4 bagian, yaitu *message flow* pada saat registrasi, pemanggilan layanan *Video on Demand*, pemutusan *Video on Demand*, dan pada saat deregistrasi.

4.2.1. Message Flow Pada Saat Registrasi

Uji coba ini dilakukan dengan meregistrasikan *client* dengan nama *deolens@open-ims.test*. Dari percobaan didapat data *message flow* sebagai berikut:

Tabel 4.1 Message Flow Registrasi

No	Source	Destination	Protocol	Info
1	Client	IMS Core	SIP	Request: REGISTER sip:open-ims.test
2	IMS Core	Client	SIP	Status: 401 Unauthorized - Challenging the UE
3	Client	IMS Core	SIP	Request: REGISTER sip:open-ims.test
4	IMS Core	Client	SIP	Status: 200 OK - SAR succesful and registrar saved
5	Client	IMS Core	SIP	Request: SUBSCRIBE sip:deolens@open-ims.test
6	IMS Core	Client	SIP	Status: 200 Subscription to REG saved
7	IMS Core	Client	SIP	Request: NOTIFY sip:deolens@192.168.1.7:5060
8	Client	IMS Core	SIP	Status: 200 OK

Dari message flow tersebut dapat dianalisa *message flow* pada saat proses registrasi adalah sebagai berikut:

1. Pada *message flow* pertama, *client* akan melakukan SIP *request* REGISTER kepada IMS Core. *Request* ini akan melalui P-CSCF dan diteruskan ke I-CSCF. I-CSCF akan menanyakan kepada HSS apakah *client* tersebut diperbolehkan untuk registrasi atau tidak. Jawaban dari HSS akan diteruskan I-CSCF ke S-CSCF. S-CSCF akan menanyakan kepada HSS informasi apa saja yang kurang dari *client* tersebut agar bisa

melakukan registrasi. Jawaban dari HSS akan dikirimkan kepada *client* dengan SIP response 401 Unauthorized.

2. Pada *message flow* kedua, response 401 Unauthorized akan dikirimkan dari IMS Core ke *client*. Response ini dikirimkan dari HSS melalui S-CSCF, I-CSCF, P-CSCF, lalu sampai di *client*.
3. Pada *message flow* ketiga, *client* akan memperbaiki SIP request REGISTER sesuai dengan response yang sebelumnya telah diterima dan mengirimkannya sesuai dengan alur *message flow* pertama namun kali ini permintaan REGISTER dari *client* diterima oleh *core*.
4. Pada *message flow* keempat, IMS Core akan mengirimkan balasan SIP 200 OK yang menandakan permintaan REGISTER dari *client* telah diterima dan *client* sudah teregistrasi.
5. Pada *message flow* kelima, *client* akan melakukan request untuk SUBSCRIBE. Request ini sebenarnya lebih banyak digunakan untuk layanan “*presence*”.
6. Pada *message flow* keenam, IMS Core akan memberi tahu bahwa request SUBSCRIBE dari *client* telah berhasil.
7. Pada *message flow* ketujuh, IMS Core akan mengirimkan notifikasi bahwa permintaan *client* telah dilaksanakan.
8. Pada *message flow* kedelapan, *client* akan membalas notifikasi dari IMS Core dengan OK.

4.2.2. Message Flow Pada Saat Pemanggilan VoD

Uji coba ini dilakukan dengan meruquest salah satu layanan VoD nama *channell@open-ims.test*. Dari percobaan didapat data *message flow* sebagai berikut:

Tabel 4.2 Message Flow Pemanggilan VoD

No	Source	Destination	Protocol	Info
1	Client	IMS Core	SIP	Request: INVITE sip:channell@iptv.open-ims.test, with session description

2	IMS Core	Application Server	SIP	Request: INVITE sip:channel1@iptv.open-ims.test, with session description
3	Application Server	IMS Core	SIP	Status: 100 Trying
4	IMS Core	Client	SIP	Status: 100 trying -- your call is important to us
5	Application Server	IMS Core	SIP	Status: 101 Dialog Establishment
6	IMS Core	Client	SIP	Status: 101 Dialog Establishment
7	Application Server	IMS Core	SIP	Status: 200 OK
8	IMS Core	Client	SIP	Status: 200 OK
9	Client	IMS Core		Request: ACK

Dari message flow tersebut dapat dianalisa *message flow* pada saat proses pemanggilan VoD adalah sebagai berikut:

1. Client akan melakukan permintaan terhadap salah satu layanan VoD dengan nama channel1@iptv.open-ims.test.
2. Permintaan dari client ini akan diteruskan oleh IMS Core ke application server yang bersangkutan.
3. Untuk memproses request dari client application server membutuhkan waktu, maka Application server akan memberitahukan kalau dia sedang mencoba memproses request dari client.
4. IMS Core akan melanjutkan response dari application server tersebut kepada client.
5. Application server akan memberitahu bahwa hubungan dengan client telah terbentuk.

6. IMS Core akan melanjutkan response dari application server tersebut kepada client.
7. Application server memberikan jawaban final dari permintaan client berupa alamat RTSP dari media sever yang dituju dalam uji coba ini alamat tersebut adalah URL="rtsp://192.168.100.5:8000/coba" di dalam pesan SIP 200 OK
8. IMS Core akan melanjutkan response dari application server tersebut kepada client.
9. Client akan mengirimkan ACK yang artinya client telah menerima respnse final dari request INVITE yang telah dia lakukan.

4.2.3. Message Flow Pada Saat Pemutusan VoD

Uji coba ini dilakukan dengan memutuskan hubungan dengan salah satu layanan VoD channel1@open-ims.test yang telah dipanggil sebelumnya. Dari percobaan didapat data *message flow* sebagai berikut:

Tabel 4.3 Message Flow Pemutusan VoD

No	Source	Destination	Protocol	Info
1	Client	IMS Core	SIP	Request: BYE
2	Application Server	IMS Core	SIP	Status: 200 OK
3	IMS Core	Client	SIP	Status: 200 OK
4	Client	IMS Core	SIP	Request: ACK

Dari message flow tersebut dapat dianalisa *message flow* pada saat proses pemutusan VoD adalah sebagai berikut:

1. Client melakukan request untuk mengakhiri layanan VoD yang sebelumnya telah dia minta dengan mengirimkan SIP request BYE kepada IMS Core.
2. Application Server membalas request dari client dengan response 200 OK

Universitas Indonesia

3. IMS Core akan melanjutkan response dari application server tersebut kepada client.
4. Client akan mengirimkan ACK yang artinya client telah menerima response final dari request yang telah dia lakukan.

4.2.4. Message Flow Pada Saat Deregistrasi

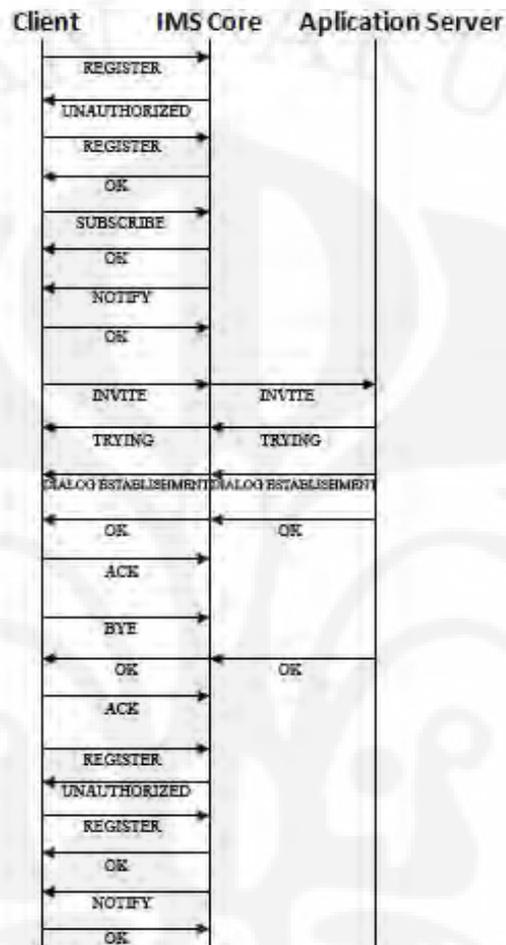
Uji coba ini dilakukan dengan melakukan proses deregistrasi *client* dengan nama *deolens@open-ims.test*. Dari percobaan didapat data *message flow* sebagai berikut:

Tabel 4.4 Message Flow Deregistrasi

No	Source	Destination	Protocol	Info
1	Client	IMS Core	SIP	Request: REGISTER sip:open-ims.test
2	IMS Core	Client	SIP	Status: 401 Unauthorized - Challenging the UE
3	Client	IMS Core	SIP	Request: REGISTER sip:open-ims.test
4	IMS Core	Client	SIP	Status: 200 OK - SAR succesful and registrar saved
5	IMS Core	Client	SIP	Request: NOTIFY sip:deolens@192.168.1.7:5060
6	Client	IMS Core	SIP	Status: 200 OK

Dari data *message flow* yang didapat pada saat deregistrasi client, terlihat bahwa *message flow* pada saat registrasi dan deregistrasi hampir sama. Hal ini karena SIP tidak mengenal request DEREGISTER, oleh karena itu pada saat register dan register request yang dilakukan sama-sama menggunakan REGISTER, yang membedakan hanyalah pada SIP di proses deregistrasi nilai “expires” adalah 0.

Secara keseluruhan *message flow* sejak registrasi sampai deregistrasi adalah sebagai berikut:



Gambar 4.2 Message Flow

4.3. ANALISA BANDWIDTH OPTIMUM

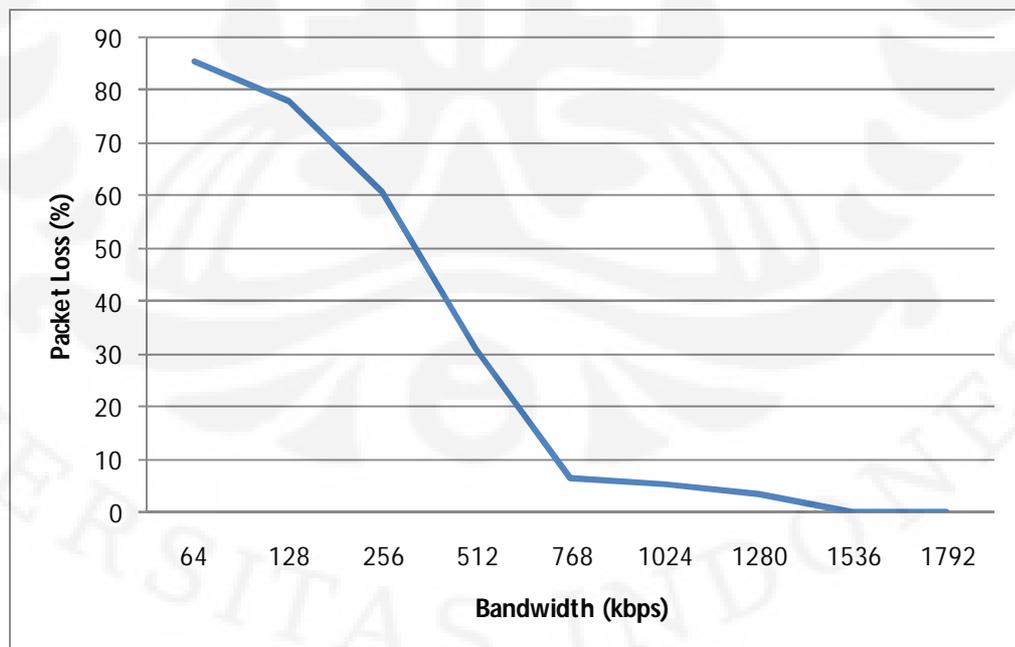
Pada uji coba ini dilakukan dengan menggunakan 1 *client* dimana nantinya *bandwidth* pada jaringan diatur dan dinaikkan mulai dari 64 kbps sampai mendapatkan *bandwidth* optimum yaitu pada saat nilai dari *packet loss* kurang dari 1% dan *delay* kurang dari 10 s sesuai dengan standard ITU-T G.1010. Uji coba ini akan dilakukan sebanyak 3 kali untuk setiap *bandwidth* dimana nantinya nilai *packet loss*, *delay*, *jitter*, dan *throughput* akan dirata-rata. Pada penelitian ini video yang digunakan adalah video dengan format MPEG-1 dengan bitrate video sebesar 1122 kbps, bitrate audio 192 kbps, dan resolusi 320 x 240 pixel. Dari percobaan yang dilakukan didapat data sebagai berikut:

Universitas Indonesia

Tabel 4.5 Hasil Pengambilan Data

Bandwidth (kbps)	Packet Loss (%)	Delay (ms)	Jitter (ms)	Troughput (packet/s)
64	85,47	234,08	175,73	5,55
128	77,87	119,43	106,84	8,38
256	60,83	56,27	72,79	17,77
512	31,03	24,16	54,61	41,4
768	6,53	15,69	48,34	63,74
1024	5,23	14,8	42,59	67,55
1280	3,27	13,83	40,46	72,32
1536	0	12,26	38,02	81,82
1792	0	12,16	38,07	86,37

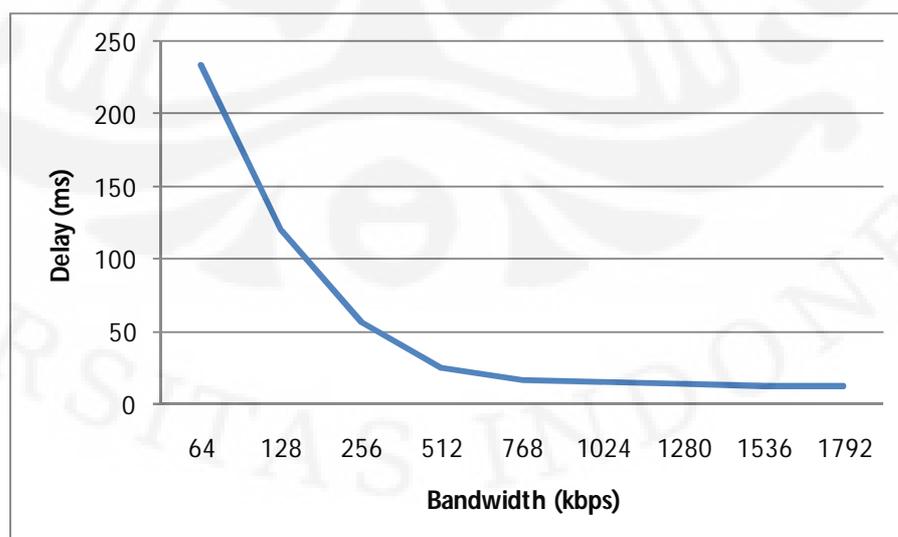
Dari data di atas jika digambarkan ke dalam grafik maka didapatkan grafik perbandingan antara *bandwidth* dan *packet loss* sebagai berikut:



Gambar 4.3 Grafik Bandwidth dan Packet Loss

Dari hasil penelitian di atas dapat dilihat bahwa pada penelitian ini terjadi packet loss atau kehilangan data. Kehilangan paket data pada proses transmisi ini terjadi ketika terdapat penumpukan data pada jalur yang dilewati pada saat beban puncak (*peak load*) yang menyebabkan kemacetan transmisi paket akibat padatnya trafik yang harus dilayani dalam batas waktu tertentu. Sehingga *frame* akan dibuang sebagaimana perlakuan terhadap *frame* data lainnya pada jaringan berbasis IP. Packet loss biasanya disebabkan oleh beberapa faktor seperti kegagalan link transmisi atau kapasitas yang tidak mencukupi. Packet loss juga dipengaruhi oleh bitrate dari codec yang digunakan. Pada bandwidth yang sama, nilai bitrate video yang besar akan meningkatkan kemungkinan terjadinya packet loss karena bitrate yang besar memerlukan bandwidth yang lebih besar pula.

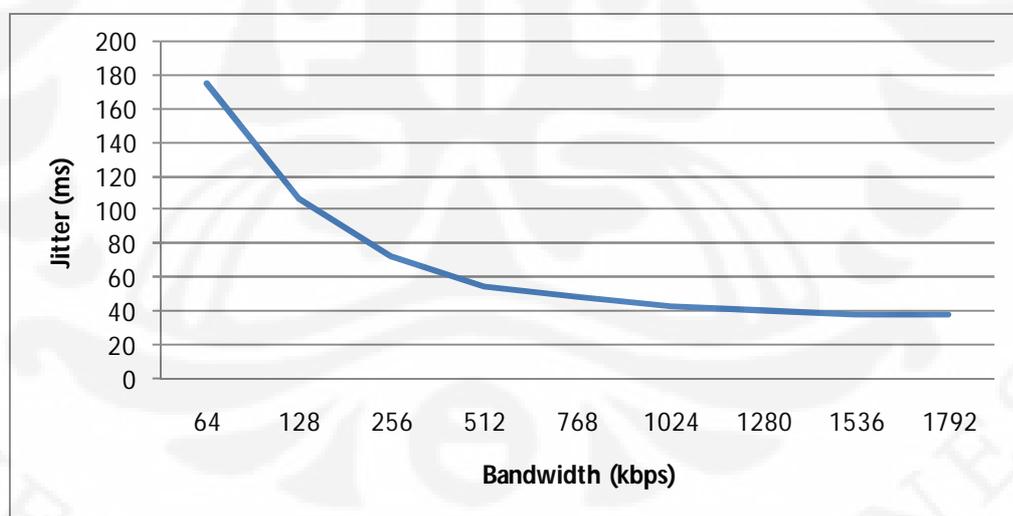
Dari data dan grafik perubahan *packet loss* terhadap *bandwidth* terlihat bahwa semakin besar *bandwidth* maka nilai *packet loss* pun menjadi semakin kecil. Hal ini dikarenakan pada saat nilai bandwidth sangat kecil maka kapasitas saluran transmisi tidak dapat memenuhi aliran trafik sehingga terjadi kemacetan transmisi dan akhirnya sebagian packet akan dibuang. Namun pada saat nilai bandwidth diperbesar maka kemacetan akan berkurang dan jumlah packet yang dibuang pun akan berkurang. Dengan menjadikan ITU-T G.1010 sebagai acuan maka nilai *packet loss* yang telah memenuhi standard *packet loss* kurang dari 1% adalah *packet loss* pada *bandwidth* di atas 1536 kbps dimana *packet loss* sudah tidak ada sama sekali.



Gambar 4.4 Grafik Bandwidth dan Delay

Dari data penelitian terlihat pada penelitian ini juga terjadi delay. Delay terjadi karena kapasitas saluran transmisi tidak dapat memenuhi beban traffic sehingga menyebabkan terjadinya kemacetan data yang mengakibatkan keterlambatan sampainya data ke tujuan. Ada beberapa penyebab terjadinya delay antara lain kongesti, kekurangan pada metode traffic shaping, penggunaan paket-paket data yang besar pada jaringan berkecepatan rendah, adanya paket-paket data dengan ukuran berbeda-beda, perubahan kecepatan antar jaringan WAN, pemadatan bandwidth secara tiba-tiba. Besarnya delay juga tergantung pada besarnya payload dari data. Semakin besar payload data maka nilai delay akan menjadi semakin besar.

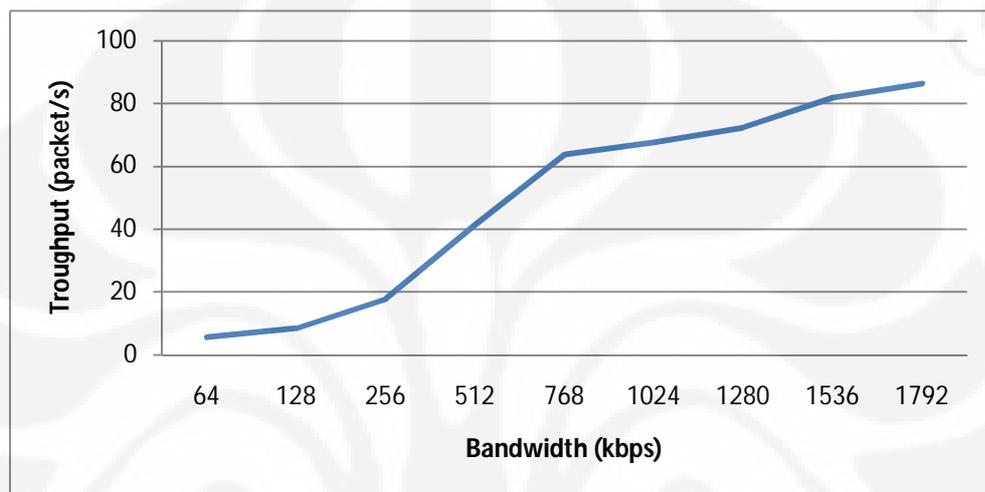
Dari data dan grafik perubahan *delay* terhadap *bandwidth* terlihat bahwa semakin besar nilai *bandwidth* maka nilai *delay* pun menjadi semakin kecil. Dengan menjadikan ITU-T G.1010 sebagai acuan dimana batas *delay* yang diperbolehkan adalah 10 s maka dari *bandwidth* terkecil yang diujicobakan pada jaringan ini yaitu 64 kbps sudah memenuhi batas delay yang diijinkan ITU-T G.1010 dimana besarnya delay adalah 234.08 ms.



Gambar 4.5 Grafik Bandwidth dan Jitter

ITU-T G.1010 hanya memberi batasan terhadap *packet loss* dan *delay*. Oleh karena itu, untuk nilai *jitter* pada penelitian ini menggunakan batasan yang dikeluarkan oleh hasil penelitian yang dibuat oleh Internet2 QoS Working Group

yang berjudul *Network QoS Needs of Advanced Internet Applications* yaitu 500 ms. Dari data dan grafik yang diperoleh terlihat bahwa nilai *jitter* akan semakin berkurang ketika nilai *bandwidth* ditingkatkan. Jika mengacu pada hasil penelitian dari Internet2 QoS Working Group tersebut maka dari *bandwidth* terkecil yang diujicobakan pada penelitian ini (64 kbps) nilai *jitter* yaitu 175.73 ms telah memenuhi standard.



Gambar 4.6 Grafik Bandwidth dan Troughput

Dari data dan grafik perubahan *troughput* terhadap *bandwidth* terlihat bahwa semakin besar *bandwidth* maka nilai *troughput* pun akan menjadi semakin besar. Ini menandakan bahwa semakin besar *bandwidth* maka semakin banyak paket yang benar yang diterima setiap detik. hal ini dikarenakan nilai *bandwidth* yang besar akan mengurangi kemungkinan hilangnya paket sehingga jumlah data yang sampai ketujuan setiap detik pun akan semakin banyak.

Nilai *troughput* juga sangat tergantung pada *bitrate* dari *codec* yang digunakan. Dengan mengasumsikan nilai *bandwidth* sudah mencukupi, maka video yang menggunakan *codec* dengan nilai *bitrate* yang lebih besar maka nilai *troughput* pun akan semakin besar karena kecepatan transfer datanya pun akan semakin besar sehingga jumlah data yang diterima setiap detik pun akan semakin besar.

Jika dilihat dari keseluruhan parameter, maka VoD ini dikatakan optimum sesuai standard ITU-T G.1010 pada *bandwidth* 1536 kbps, yaitu dengan packet

Universitas Indonesia

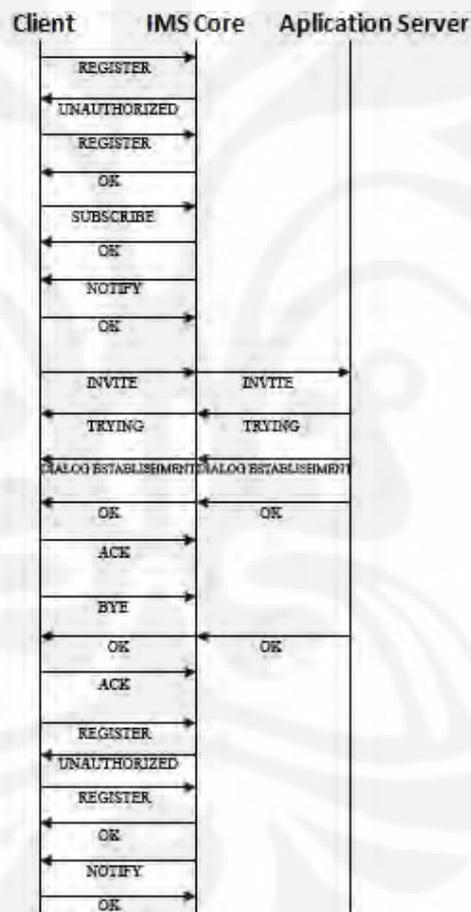
loss 0 % dan delay 12.26 ms. Namun hasil penelitian ini hanya terbatas pada VoD dengan format video MPEG-1 dengan bitrate video sebesar 1122 kbps, bitrate audio 192 kbps, dan resolusi 320 x 240 pixel. Apabila digunakan video dengan bitrate yang lebih tinggi maka bandwidth yang dibutuhkan pun akan semakin besar. Selain itu penelitian ini hanya mencari bandwidth optimum untuk 1 client saja. Apabila lebih dari 1 client yang mengakses layanan Video on Demand ini maka kapasitas bandwidth yang dibutuhkan pun akan semakin besar.

Jika dibandingkan dengan layanan IPTV dengan dasar yang sama yaitu Open IMS, maka layanan Video on demand ini memerlukan bandwidth yang lebih kecil dari IPTV. Hal ini disebabkan karena pada layanan IPTV codec audio yang digunakan adalah MP2. Bitrate dari codec MP2 yang digunakan untuk IPTV adalah 384 kbps, sementara untuk layanan Video on Demand pada penelitian ini codec audio yang digunakan adalah MPEG-1 dengan bitrate sebesar 192 kbps. Karena bitrate audio IPTV lebih besar dari bitrate audio dari video on Demand maka IPTV memerlukan bandwidth yang lebih besar dari Video on Demand.

BAB 5

KESIMPULAN

1. Jaringan Open IMS yang terdiri dari HSS, P-CSCF, I-CSCF, S-CSCF dan ditambah Application Server dan Media Server sudah mendukung untuk layanan *Video on Demand*.
2. Pada layanan *Video on Demand* berbasis open IMS ini *message flow* dari mulai proses registrasi, pemanggilan layanan VoD, pemutusan layanan VoD, sampai proses deregistrasi adalah sebagai berikut:



3. Layanan *Video on Demand* berbasis open IMS dimana video yang distreamingkan memiliki format MPEG-1 dengan bitrate video sebesar 1122 kbps, bitrate audio 192 kbps, dan resolusi 320 x 240 pixel ini dapat berjalan secara optimum pada *bandwidth* minimum sebesar 1536 kbps dengan *packet loss* 0% dan *delay* 12.26 ms.

DAFTAR REFERENSI

- [1] Hens, F.J & Caballero, J.M.(2008). *Triple Play* (hal 281). John Wiley & Sons
- [2] Dewi Asri (2009, Juli). *Rancang Bangun Dan Unjuk Kerja Qos Objektif Voip Berbasis Sip Menggunakan Codec G-711 M-Law Dan GSM 06.10*, Skripsi. Depok: Universitas Indonesia.
- [3] Hens, F.J & Caballero, J.M.(2008). *Triple Play* (hal 109). John Wiley & Sons
- [4] *SIP (Session Initiation Protocol)*. Diakses 23 April 2010, Gammacute's Blog <http://gammacute.wordpress.com/2009/08/>
- [5] *Protocol RTSP*. Diakses 23 April 2010, dari perpustakaan IT Telkom <http://www.ittelkom.ac.id/library/sip-session-initiation-protocol>
- [6] Hens, F.J & Caballero, J.M.(2008). *Triple Play* (hal 121). John Wiley & Sons
- [7] M.Handley, V.Jacobson,& C.Perkins.*SDP:Session Initiation Protocol*. July 2006.RFC 4566.
- [8] Dewi Asri (2009, Juli). *Rancang Bangun Dan Unjuk Kerja Qos Objektif Voip Berbasis Sip Menggunakan Codec G-711 M-Law Dan GSM 06.10*, Skripsi. Depok: Universitas Indonesia.