



**UNIVERSITAS INDONESIA**

**ANALISA IMPLEMENTASI IPTV DENGAN  
MENGUNAKAN OPEN IMS CORE**

**SKRIPSI**

**HARTONO  
06 06 07 3953**

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



**UNIVERSITAS INDONESIA**

**ANALISA IMPLEMENTASI IPTV DENGAN  
MENGUNAKAN OPEN IMS CORE**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**HARTONO  
06 06 07 3953**

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

## HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : Hartono**  
**NPM : 0606073953**  
**Tanda Tangan :**  
**Tanggal : 28 Juni 2010**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Hartono  
NPM : 0606073953  
Program Studi : Teknik Elektro  
Judul Skripsi : Analisa Implementasi IPTV Dengan Menggunakan  
Open IMS Core

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 : Ir. Muhammad Asvial, M.Sc, Ph.D

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

Ditetapkan di : Depok

Tanggal : 28 Juni 2010

## UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini tepat pada waktunya. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Dalam proses penyusunan skripsi ini, saya mendapat banyak bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Ir. Djamhari Sirat M.Sc, Ph.D, selaku pembimbing yang telah meluangkan waktu dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Bapak Nooriza (Depkominfo), bapak Yulianus (Indosat), bapak Panca (Telkomsel), dan bapak Bambang (Telkom) atas segala bantuan dan masukan dalam penyusunan skripsi ini;
3. Rekan-rekan satu bimbingan: Deolens Manurung, Mark Swapo, R. Kharisma, dan Zaneta Pelangi atas kerjasamanya selama pengerjaan skripsi ini;
4. Orang tua dan keluarga yang telah member dukungan material dan moral; dan
5. Sahabat dan rekan mahasiswa yang telah ikut membantu memberi segala dukungan.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 28 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 : Hartono  
NPM : 0606073953  
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 IMPLEMENTASI IPTV DENGAN MENGGUNAKAN OPEN IMS  
CORE**

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

(Hartono)

## ABSTRAK

Nama : Hartono  
Program Studi : Teknik Elektro  
Judul : Analisa Implementasi IPTV Dengan Menggunakan Open IMS Core

Skripsi ini membahas mengenai salah satu aplikasi dari *IP Multimedia Subsystem* (IMS) yakni *Internet Protocol Television* (IPTV). Dalam emulasi jaringan ini digunakan Open IMS Core, UCT IMS Client, dan jaringan yang terdiri dari 2 *router*, dan 1 buah *switch*. Sedangkan untuk pengambilan dan pengujian data digunakan program *Wireshark* untuk menganalisa *message flow* registrasi *client*, permohonan layanan IPTV sampai deregistrasi dan parameter-parameter *Quality of Service* (QoS) seperti: *delay*, dan *packet loss*.

Pengujian dilakukan dengan merubah *bandwidth* pada hubungan kedua rouer, dengan kisaran nilai dari 64 kbps sampai 2.112 kbps. Pada setiap nilai *bandwidth*, dilakukan 3 kali percobaan dan kemudian nilai parameter-parameter QoS yang didapatkan akan dirata-ratakan. Hasil dari pengujian ini didapatkan *bandwidth* optimum untuk layanan iPTV yakni 2112 kbps dengan *packet loss* 0,85% (sesuai dengan standar ITU-T yakni kurang dari 1 %).

Kata Kunci:

IPTV, Open IMS Core, *message flow*, *delay*, dan *packet loss*

## ABSTRACT

Name : Hartono  
Study Program : Teknik Elektro  
Title : Analysis on the Implementation of IPTV Using the Open IMS Core

This final assignment discusses about one of the applications of IP Multimedia Subsystem (IMS), namely Internet Protocol Television (IPTV). This network emulation is using Open IMS Core, UCT IMS Client, and network that consisting of two router and a switch. While for the taking and testing data are used Wireshark program to analyze the message flow from client registration, request for IPTV service, client deregistration, and Quality of Service (QoS) parameters such as: delay, and packet loss.

Testing is doing by changing the bandwidth on the network between two router, with a range of values from 64 kbps to 2,112 kbps. In each bandwidth, there are three times experiments and then the QoS parameters that obtainable is divided to get mean score. The result from this test showed the optimum bandwidth for IPTV service is 2112 kbps with packet loss 0.85 % (in accordance with ITU-T standard that is less than 1%).

Keywords:

IPTV, Open IMS Core, message flow, delay, and packet loss



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
UCAPAN TERIMA KASIH .....	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 .....	1
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	2
1.5 Metode Penelitian .....	2
1.6 Sistematika Penulisan .....	3
2. LANDASAN TEORI .....	5
2.1 IP Multimedia Subsystem (IMS) .....	5
2.1.1 Home Subscriber Server (HSS) .....	7
2.1.2 Proxy Call Session Control Function (P-CSCF) .....	7
2.1.3 Interrogating Call Session Control Function (P-CSCF) .....	8
2.1.4 Serving Call Session Control Function (P-CSCF) .....	8
2.2 IPTV .....	8
2.3 Protokol-Protokol Pada IPTV .....	8
2.3.1 Real-Time Transport Protocol (RTP) .....	10
2.3.2 Real-Time Transport Control Protocol (RTCP) .....	10
2.3.3 Session Initiation Protocol (SIP) .....	10
2.3.4 Session Description Protocol (SDP) .....	13
2.4 Parameter-Parameter QoS .....	14
2.4.1 Delay .....	14
2.4.2 Packet Loss .....	15
3. PERANCANGAN DAN KONFIGURASI IPTV BERBASIS OPEN IMS ...	16
3.1 Open IMS Core .....	16
3.1.1 Instalasi dan Konfigurasi Open IMS Core .....	16
3.1.2 Mengubah Open IMS Core ke Spesifik IP .....	21

3.2 Application layer IPTV .....	25
3.2.1 Instalasi Application Server IPTV .....	25
3.2.2 Konfigurasi Application Server IPTV.....	25
3.3 IMS Client .....	28
3.3.1 Instalasi dan konfigurasi IMS Client.....	28
3.3.2 Membuat Account Client .....	30
3.4 Topologi Jaringan .....	33
4. HASIL PENGUKURAN DAN ANALISA PARAMETER QoS .....	36
4.1 Message Flow .....	36
4.2 Pengambilan dan Pengolahan Data.....	38
4.3 Analisa Parameter QoS .....	42
5. KESIMPULAN.....	44
DAFTAR REFERENSI .....	45

## DAFTAR TABEL

Tabel 2.1 Port Number Elemen OpenIMSCore .....	22
Tabel 4.1 Message Flow .....	36
Tabel 4.2 Packet Loss dan Delay Pada Bandwidth 64 kbps .....	39
Tabel 4.3 Packet Loss dan Delay Pada Bandwidth 128 kbps .....	39
Tabel 4.4 Packet Loss dan Delay Pada Bandwidth 256 kbps .....	39
Tabel 4.5 Packet Loss dan Delay Pada Bandwidth 512 kbps .....	39
Tabel 4.6 Packet Loss dan Delay Pada Bandwidth 768 kbps .....	40
Tabel 4.7 Packet Loss dan Delay Pada Bandwidth 1024 kbps .....	40
Tabel 4.8 Packet Loss dan Delay Pada Bandwidth 1280 kbps .....	40
Tabel 4.9 Packet Loss dan Delay Pada Bandwidth 1536 kbps .....	40
Tabel 4.10 Packet Loss dan Delay Pada Bandwidth 1792 kbps .....	41
Tabel 4.11 Packet Loss dan Delay Pada Bandwidth 2048 kbps .....	41
Tabel 4.12 Packet Loss dan Delay Pada Bandwidth 2112 kbps .....	41

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Tentang IMS .....	5
Gambar 2.2 Open IMS Core Pada Jaringan IMS .....	6
Gambar 2.3 Elemen OpenIMSCore .....	7
Gambar 3.1 Edit Named.conf .....	18
Gambar 3.2 Edit Resolv.conf .....	18
Gambar 3.3 Edit Hosts .....	19
Gambar 3.4 Running PCSCF .....	20
Gambar 3.5 Running ICSCF .....	21
Gambar 3.6 Running SCSCF .....	21
Gambar 3.7 Running FHoSS .....	22
Gambar 3.8 Edit Resolv.conf ke Spesifik IP .....	23
Gambar 3.9 Edit Hosts ke Spesifik IP .....	23
Gambar 3.10 Edit Open-ims.dnszone ke Spesifik IP .....	24
Gambar 3.11 FHoSS .....	25
Gambar 3.12 Application Server IPTV .....	26
Gambar 3.13 Trigger Point IPTV .....	26
Gambar 3.14 Initial Filter Criteria IPTV.....	27
Gambar 3.15 Shared iFC Sets IPTV .....	27
Gambar 3.16 Service Profile IPTV .....	28
Gambar 3.17 UCTIMSCLIENT .....	29
Gambar 3.18 Registrasi UCTIMSCLIENT .....	29
Gambar 3.19 Tampilan IPTV .....	30
Gambar 3.20 IMS Subscription .....	30
Gambar 3.21 Bagian Atas Private User Identity –IMPI- .....	31
Gambar 3.22 Bagian Bawah Private User Identity –IMPI- .....	31
Gambar 3.23 Public User Identity –IMPU- .....	32
Gambar 3.24 UCTIMSCLIENT Preferences .....	32
Gambar 3.25 Topologi Jaringan .....	33
Gambar 4.1 Grafik Perbandingan Bandwidth dengan Packet Loss .....	42
Gambar 4.2 Grafik Perbandingan Bandwidth dengan Delay .....	42

# BAB 1

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Teknologi telekomunikasi pada saat ini berkembang dengan pesat sehingga hal ini telah menyebabkan para *vendor* dan *operator* untuk saling bersaing dalam mengembangkan teknologi-teknologi baru yang diharapkan bisa meningkatkan kualitas dari layanan yang akan ditawarkan kepada konsumennya. Salah satu teknologi yang sedang hangat dalam tahap pengembangan adalah *IP Multimedia Subsystem* (IMS).

IMS merupakan arsitektur jaringan telekomunikasi berbasis multimedia IP (*internet protocol*) yang melingkupi layanan suara, video, *Internet Protocol Television* (IPTV) dan data. Selain itu, IMS juga memberikan kemudahan-kemudahan dalam mengakses layanan-layanan di atas dengan hanya bermodalkan koneksi internet.

Salah satu aplikasi IMS yang berpotensi dan menarik untuk diterapkan adalah IPTV. IPTV menyediakan saluran televisi yang tidak terbatas, kualitas gambar dan suara yang lebih bagus serta dapat diakses melalui jaringan kabel telepon yang sudah ada maupun *mobile* yakni menggunakan jaringan *wireless*.

Oleh karena itu, maka skripsi ini bertujuan untuk memahami cara kerja dari layanan IPTV pada IMS dengan cara mengemulasikan jaringan IPTV ini dalam suatu *open source* yaitu Open IMS Core pada *Operating System* (OS) Linux. Pada bagian analisis akan difokuskan pada *messages flow* dari layanan IPTV beserta dengan parameter-parameter *Quality of Service* (QoS) untuk mengetahui apakah layanan ini sudah layak untuk diimplementasikan.

### 1.2 PERUMUSAN MASALAH

Dalam skripsi ini, akan diemulasikan jaringan IPTV dalam Open IMS Core pada *Operating System* Linux. Sehingga terdapat beberapa rumusan masalah yang akan diperhatikan dan dianalisa pada skripsi ini, yaitu:

1. Bagaimana cara kerja dan konsep dari Open IMS Core yang digunakan untuk menjalankan aplikasi IPTV ?

2. Bagaimana *message flow* pada saat *client* melakukan proses registrasi, pemanggilan layanan IPTV, pemutusan layanan IPTV sampai proses deregistrasi ?
3. Berapa nilai dari parameter-parameter QoS pada saat *client* memanggil layanan IPTV ?

### 1.3 TUJUAN PENELITIAN

Tujuan dari penyusunan dan pembuatan skripsi ini adalah :

1. Mengemulasikan jaringan IPTV dengan menggunakan Open IMS Core
2. Memahami cara kerja dan konsep dari Open IMS Core yang digunakan untuk menjalankan aplikasi IPTV
3. Memahami *message flow* pada saat *client* melakukan registrasi, pemanggilan layanan IPTV, pemutusan layanan IPTV sampai proses deregistrasi
4. Menganalisa parameter-parameter QoS pada saat *client* memanggil layanan IPTV

### 1.4 BATASAN MASALAH

Pada skripsi ini, masalah akan dibatasi pada pengemulasian jaringan IPTV dengan menggunakan Open IMS Core, analisa *message flow* dan parameter-parameter QoS (*delay* dan *packet lost*) saat pemanggilan layanan IPTV.

### 1.5 METODE PENELITIAN

Metode penelitian yang digunakan dalam penyusunan skripsi ini adalah:

1. Tinjauan Pustaka, yaitu dengan melakukan studi literatur dari buku-buku pustaka, *website*, dan modul-modul yang berkaitan dengan skripsi ini
2. Diskusi dan tanya jawab dengan dosen pembimbing dan orang-orang yang ahli pada bidang yang berkaitan dengan skripsi ini
3. Perancangan perangkat lunak
4. Pengujian dan pengambilan data
5. Analisa data

Penelitian dilakukan melalui prosedur sebagai berikut:

1. Penginstalan dan konfigurasi Open IMS Core dan Application Server IPTV pada komputer *server*
2. Penginstalan IMS Client pada komputer *client*
3. Pembangunan skenario jaringan untuk penelitian
4. Registrasi IMS Client pada IMS Core
5. *Request* dan inisiasi salah satu channel IPTV di IMS Client
6. Pengukuran QoS jaringan IMS menggunakan Wireshark
7. Analisa *message flow* dan hasil pengukuran QoS

### 1.6 SISTEMATIKA PENULISAN

Pada skripsi ini terdapat 5 bab di mana masing-masing bab akan dijelaskan tentang:

1. Bab I : PENDAHULUAN

Pada bab ini, akan dijelaskan mengenai latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metode penelitian dan sistematika penulisan.

2. Bab II : LANDASAN TEORI

Pada bab ini, akan dijelaskan landasan teori tentang *IP Multimedia Subsystem* (IMS), IPTV dan protokol-protokolnya serta parameter-parameter QoS.

3. Bab III : PERANCANGAN DAN KONFIGURASI IPTV BERBASIS OPEN IMS

Pada bab ini, akan dijelaskan tentang tahap-tahap instalasi dan konfigurasi OpenIMSCore, UCT IMS Client, dan *Application Server* IPTV.

4. Bab IV : HASIL PENGUKURAN DAN ANALISA PARAMETER QOS

Pada bab ini, akan dibahas tentang analisa *message flow* pada jaringan IPTV dan pengukuran parameter-parameter QoS-nya.

5. Bab V : KESIMPULAN

Pada bab ini, akan dijelaskan tentang kesimpulan yang dapat diambil dari penelitian ini.





## BAB 2

### LANDASAN TEORI

Pada saat ini, komunikasi selular dan internet sudah menjadi hal yang lumrah dalam kehidupan kita sehari-hari. Kedua paradigma sukses dalam komunikasi ini telah menjadi fokus utama dari jaringan telekomunikasi saat ini, yang secara bersamaan telah didukung oleh sistem IMS (*IP Multimedia Subsystem*). IMS memungkinkan tersedianya akses seluler di manapun ke seluruh layanan internet yang tersedia sehingga kita dapat *browsing* website, membaca email, menonton film atau ikut *video conference* di manapun anda berada.

#### 2.1 IP Multimedia Subsystem (IMS)

IMS merupakan komponen utama untuk mendukung konvergensi layanan di mana teknologi ini didesain untuk mengisi *gap* antara telekomunikasi tradisional dan teknologi internet. IMS dirancang untuk memberikan sejumlah fungsi yang dibutuhkan untuk mendukung layanan berbasis IP melalui jaringan bergerak dan nirkabel. IMS secara spesifik dirancang untuk mendukung layanan multimedia yang bersifat *real-time* seperti *video telephony*, *video conference*, dan *push services*.



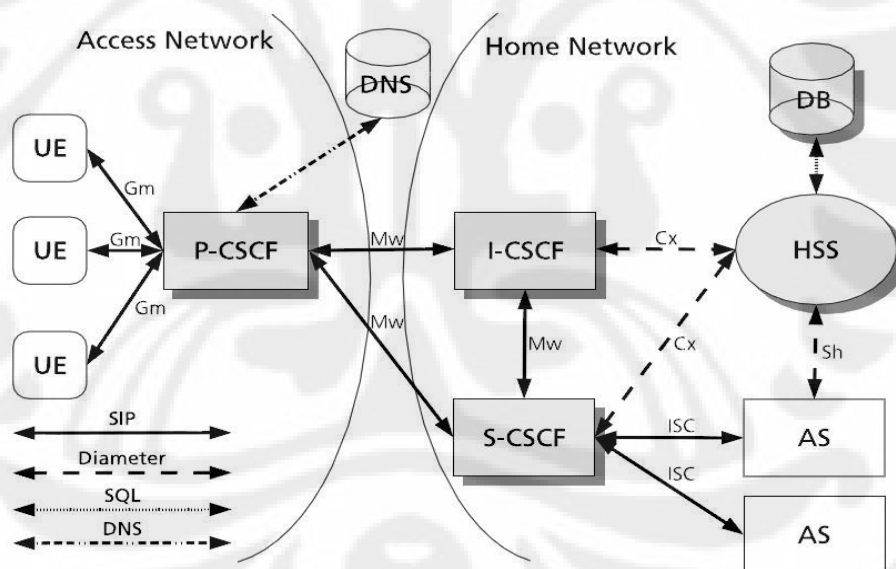
Gambar 2.1 Ilustrasi Tentang IMS

Sumber: <http://joudane.wordpress.com/2008/07/15/ims-ip-multimedia-subsystem/>

Dalam dunia kovergensi, IMS memainkan peran penting dimana dengan bantuan SIP, IMS dapat memberikan layanan berbasis IP seperti VoIP, *video*

*conference*, IPTV, dan layanan multimedia lainnya. SIP merupakan salah satu *signaling protocol* pada IMS dimana SIP memfasilitasi interkoneksi antara jaringan *fixed* dan jaringan bergerak.

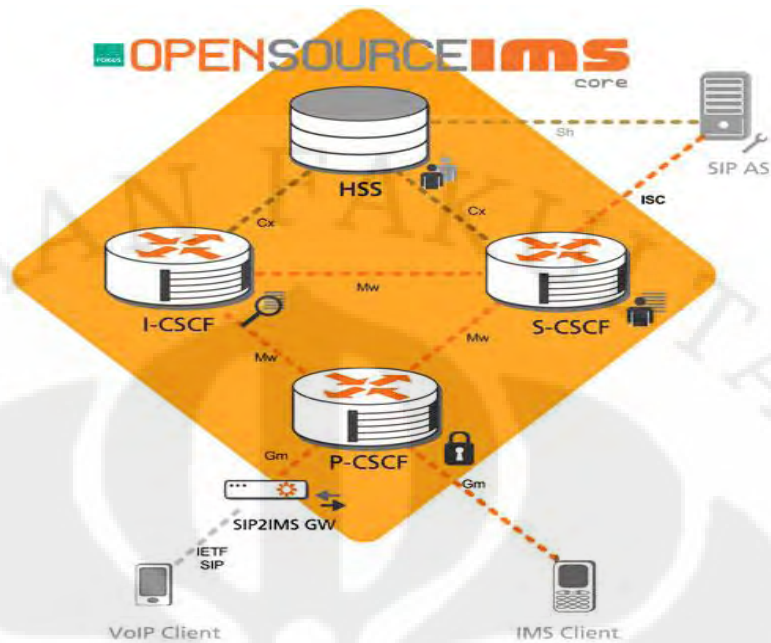
Latar belakang dikembangkan teknologi IMS ini adalah kemudahan dalam pembuatan layanan (*service*) baru pada jaringan telekomunikasi khususnya layanan multimedia. Selain itu, IMS juga memberikan kemudahan dalam integrasi dengan internet dan meningkatkan kecepatan dalam mengakses data. Teknologi IMS menggunakan SIP (*Session Initiation Protocol*) sebagai protokol pengontrol sesi (*session control*) yang dilakukan oleh pengguna yang satu dengan yang lainnya ataupun dengan suatu aplikasi. Untuk menunjang fungsinya sebagai *session control*, maka IMS menyediakan beberapa fungsi seperti mekanisme *charging*, *subscriber profile management* dan pengalokasian *Quality of Service* (QoS) pada media transmisinya.



Gambar 2.2 Open IMS Core Pada Jaringan IMS

Sumber: *Design and Implementation of an Open IMS Core*

Bagian terpenting dari jaringan IMS terletak pada IMS Core-nya yang terdiri dari beberapa komponen yaitu: *Home Subscriber Server* (HSS), *Proxy CSCF* (P-CSCF), *Interrogating CSCF* (I-CSCF), dan *Serving CSCF* (S-CSCF). [2]



Gambar 2.3 Elemen OpenIMSCore

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

#### 2.1.1 Home Subscriber Server (HSS)

Pada implementasinya nanti, setiap pelanggan IMS akan memiliki satu atau lebih identitas pribadi (*Private User Identity*). Identitas ini berisi tentang informasi akan layanan-layanan apa saja yang bisa di akses oleh pengguna, berlaku secara global dan identitas ini akan disimpan di dalam *Home Subscriber Server* (HSS) ketika kita melakukan registrasi.

HSS ini dapat dipandang sebagai evolusi dari *Home Location Register* (HLR) yang berfungsi sebagai database informasi tentang para pengguna. HSS menyimpan profil pengguna, informasi tentang pengguna mana saja yang sedang teregister serta status lokasi dari semua pengguna. Informasi-informasi ini diperlukan pada saat *Authentication*, *Authorization*, dan *Accounting* (AAA) dimana S-CSCF akan mengakses ke HSS yang keduanya terhubung dengan *Cx reference point*.

#### 2.1.2 Proxy Call Session Control Function (P-CSCF)

Ketika kita melakukan akses layanan IMS maka titik masuk pertama menuju IMS adalah P-CSCF. P-CSCF berperan dalam meneruskan SIP messages antara *user equipment* dengan node yang benar pada IMS. Bila pengakses layanan berada di luar *server* asalnya maka *user equipment* akan melakukan akses ke P-

CSCF jaringan tempat pengakses layanan berada dan kemudian diteruskan ke IMS *server* asalnya.

### 2.1.3 Interrogating Call Session Control Function (I-CSCF)

I-CSCF merupakan titik kontak untuk pengguna di jaringan asalnya dimana I-CSCF ini berfungsi dalam menentukan S-CSCF yang tepat untuk pengguna dengan berdasarkan informasi-informasi seperti: lokasi, kapabilitas dan availabilitas S-CSCF yang ada. Setelah memilih S-CSCF yang sesuai maka I-CSCF akan menginformasikan S-CSCF yang harus dipilih oleh pengguna berdasarkan data di HSS.

### 2.1.4 Serving Call Session Control Function (S-CSCF)

S-CSCF berfungsi dalam menyediakan proses registrasi, *back-2-back agent* untuk kontrol panggilan, dan layanan bagi pengguna sehingga pengguna dapat mengakses aplikasi-aplikasi yang disediakan. Selain itu, S-CSCF juga berfungsi dalam mentranslasi format dari perangkat telekomunikasi seperti telepon rumah dan telepon seluler menjadi SIP URI dengan mekanisme translasi ENUM DNS.

## 2.2 IPTV

Berdasarkan definisi menurut ITU-T, IPTV adalah teknologi yang merupakan konvergensi antara layanan *broadcasting* dan layanan telekomunikasi dimana melalui QoS terkontrol, konten multimedia seperti video, audio, dan data dapat dikirimkan ke pelanggan yang memiliki divais televisi dan *set top box*, melalui system komunikasi berbasis internet baik melalui media *wire* ataupun *wireless*.

IPTV merupakan sistem transmisi televisi digital menggunakan IP (*Internet Protocol*) yang melewati infrastruktur jaringan IP dengan pita lebar (*broadband*). Pita lebar dibutuhkan untuk mengirimkan format gambar bergerak dengan kualitas yang baik dan *real time*. Layanan ini lebih sering ditawarkan bersamaan dengan layanan internet dan *Voice over IP* (VoIP) yang disediakan oleh *provider* jaringan telekomunikasi.

Sistem ini yang digunakan untuk mengirim layanan televisi digital kepada konsumen yang terdaftar (sebagai *subscriber*) dalam sistem tersebut. Pengiriman (sinyal) digital televisi tersebut memungkinkan diselenggarakan dengan menggunakan *Internet Protocol* melewati sebuah koneksi *broadband*, biasanya digunakan dalam sebuah jaringan yang termanajemen dengan baik daripada internet publik dengan tujuan agar kualitas pelayanan terjamin. Kebanyakan layanan ini disediakan bersama dengan permintaan fasilitas video. IPTV pada dasarnya dibuat dari kombinasi antara 2 elemen penting, yaitu protokol-protokol IP dan *digital television*: [3]

1. Protokol-protokol IP bertanggung jawab pada pemaketan sinyal TV yang akan dirutekan dari *broadcaster* ke sisi pelanggan. Protocol IP juga menyediakan interaksi antara pelanggan, jaringan, layanan, dan penyedia layanan
2. *Digital television* bertanggung jawab untuk masalah audio, video, kompresi data, dan format transmisi sesuai dengan standard, seperti MPEG, yang mengatur resolusi image dan pengaturan program.

Beberapa *feature* yang ditawarkan oleh IPTV adalah:

1. IPTV dapat menyiarkan secara *live* program-program televisi seperti: film, *talkshow*, dan sebagainya.
2. IPTV dapat memberikan layanan yang ekonomis namun dengan tidak mengorbankan kualitas layanan. Ini karena teknologi *bandwidth* transmisi yang efisien.
3. IPTV mendukung format standard MPEG (*Motion Picture Experts Group*) untuk memberikan *high quality, full motion video*. *Feature* ini merupakan tambahan terhadap standard CODEC (*compression/decompression*) untuk menjamin kualitas gambar yang optimal sesuai dengan spesifikasi aplikasi dan *bandwidth* yang tersedia.
4. Bila dibandingkan dengan metode tutorial yang konvensional, IPTV lebih efisien karena tidak perlu membayar instruktur, biaya print materi relatif lebih sedikit, tidak perlu menyewa ruang khusus (karena IPTV dapat diakses oleh setiap meja selama terkoneksi dalam satu LAN/WAN).

## 2.3 Protokol-Protokol Pada IPTV

### 2.3.1 Real-Time Transport Protocol (RTP)

RTP merupakan sebuah protokol lapisan *transport* yang mampu mentransmisikan data secara *real time* [1]. RTP biasanya digunakan dalam proses *streaming* video dan audio, di mana besarnya paket video dan audio ini bergantung pada *codec* yang digunakan. Jika paket RTP hilang pada jaringan maka RTP tidak akan melakukan *retransmission* sehingga *user* tidak perlu untuk menunggu terlalu lama atau dengan kata lain *delay* yang terjadi kecil karena tidak diperlukan permintaan *retransmission*.

Selain itu, RTP juga berfungsi dalam menyediakan layanan penyampaian *end to end* untuk data yang bersifat *real time*. Dalam implementasinya, RTP biasanya digunakan secara bersamaan dengan protokol UDP karena penggunaannya yang bersifat *real time* sehingga protokol ini tidak menjamin adanya QoS untuk layanannya. Dalam pengaturan QoS untuk layanan seperti IPTV pada IMS dapat digunakan *Policy and Charging Rule Function* (PCRF), namun pada skripsi ini tidak akan membahasnya.

### 2.3.2 Real-Time Transport Control Protocol (RTCP)

RTCP merupakan suatu protokol yang biasanya dipakai bersama-sama dengan RTP. RTCP mempunyai 2 komponen penting pada paketnya, yaitu: *sender report* dan *receiver report*. Pada *sender report* terdapat informasi tentang berapa banyak data yang dikirimkan, pengecekan *timestamp* pada header RTP dan pengecekan tiap data sesuai dengan *timestamp*-nya masing-masing. Sedangkan pada *receiver report*, terdapat informasi jumlah paket yang hilang selama terjadi pengiriman data, menampilkan *timestamp* terakhir dan *delay* sejak pengiriman *sender report* yang terakhir.

### 2.3.3 Session Initiation Protocol (SIP)

*Session Initiation Protocol* (SIP) merupakan suatu *signaling protocol* yang terletak pada *layer* aplikasi yang berfungsi dalam membangun, memodifikasi, dan mengakhiri suatu sesi multimedia yang bisa saja berupa text, suara maupun video. Standar protokol multimedia ini pertama kali dikeluarkan oleh grup yang tergabung dalam *Multiparty Multimedia Session Control* (MMUSIC) yang berada

dalam organisasi *Internet Engineering Task Force* (IETF) yang didokumentasikan ke dalam dokumen *Request For Command* (RFC) 2543 pada bulan maret 1999.

SIP tidak menyediakan layanan secara langsung, tetapi cuma menyediakan struktur yang dapat digunakan oleh protokol aplikasi lainnya untuk memberikan layanan yang lebih lengkap bagi pengguna, misalnya dengan RTP (*Real Time Transport Protocol*) untuk transfer data secara *real-time*, dengan SDP (*Session Description Protocol*) untuk mendeskripsikan sesi multimedia, dan dengan MEGACO (*Media Gateway Control Protocol*) untuk komunikasi dengan PSTN (*Public Switch Telephone Network*).

Meskipun demikian, fungsi dan operasi dasar SIP tidak tergantung pada protokol tersebut. SIP juga tidak tergantung pada protokol *layer transport* yang digunakan. Pembangunan suatu komunikasi multimedia dengan SIP dilakukan melalui beberapa tahap :

1. *User Location* adalah menentukan lokasi pengguna yang akan berkomunikasi.
2. *User Availability* adalah menentukan ketersediaan pihak yang dipanggil untuk terlibat dalam komunikasi.
3. *User Capability* adalah menentukan media maupun parameter yang berhubungan dengan media yang digunakan untuk komunikasi.
4. *Session Setup* adalah pembentukan hubungan antara pihak pemanggil dengan pihak yang dipanggil.
5. *Session management* yaitu meliputi transfer, modifikasi, dan pemutusan sesi. Secara garis besar SIP merupakan protokol yang digunakan dalam untuk membangun, memodifikasi, dan mengakhiri suatu sesi. Penggunaan protokol codec video , audio dan *Real-time Protocol* dengan H.323 tetap sama, hanya berbeda dalam sesi *signaling* sambungan VoIP.

SIP dapat dikatakan bersifat *client-server*, yang berarti *client* akan mengirimkan *request* ke *server*. Kemudian, *server* mengolah *request* dan memberikan *response* terhadap *request* tersebut ke *client*. *Request* dan *responses* terhadap *request* disebut transaksi SIP. Ada enam tipe dari request messages :

1. INVITE : menunjukkan bahwa pengguna atau *service* sedang diundang untuk bergabung dalam sesi multimedia. Isi dari pesan ini akan memasukkan suatu uraian menyangkut sesi untuk pengguna yang akan diundang.
2. ACK : mengkonfirmasi bahwa *client* telah menerima suatu *final response* untuk suatu INVITE *request* , dan hanya digunakan di INVITE *request*.
3. OPTION : digunakan untuk *query* suatu *server* tentang kemampuan yang dimilikinya.
4. BYE : dikirim oleh *user agent client* untuk menunjukkan pada *server* bahwa sesi ingin segera diakhiri.
5. CANCEL : digunakan untuk membatalkan suatu *request* yang sedang menunggu keputusan.
6. REGISTER : digunakan oleh *client* untuk mendaftarkan informasi tentang identitas pribadi.

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

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

Berikut ini adalah empat server yang termasuk dalam SIP server beserta dengan fungsinya masing-masing:



#### 1. *Proxy server*

Server ini berperan sebagai perantara dalam melakukan *request* dari *client* ke *server* dan melakukan *response* dari *server* ke *client*. Selain itu, *proxy server* juga berfungsi untuk melakukan *routing*, memastikan *request* disampaikan pada pengguna yang tepat, dan juga membuat kebijakan seperti menyakinkan bahwa pengguna tertentu yang diijinkan untuk melakukan panggilan.

#### 2. *Redirect Server*

*Server* ini merupakan kesatuan logika yang mengarahkan satu klien pada perangkat pengganti dari *Uniform Resource indicators* (URIs) untuk menyelesaikan tugas *request*.

#### 3. *Registrar Server*

*Server* ini berperan dalam menerima dan memproses pesan pendaftaran yang mengijinkan lokasi dari suatu *endpoint* dapat diketahui keberadaannya. *Registrar Server* ini kerjanya berhubungan dengan *Location Server*.

#### 4. *Location Server*

*Server* ini menyediakan *service* untuk *database* abstrak yang berfungsi mentranslasikan alamat dengan kata / keterangan yang ada pada domain jaringan.

### 2.3.4 Session Description Protocol (SDP)

Dalam melakukan *session establishment*, SIP tidak mendeskripsikan sesi multimedia yang akan dibangun, namun fungsi tersebut dijalankan oleh protokol lain yaitu *session description protocol* (SDP). SDP adalah protokol layer aplikasi yang berfungsi untuk memberikan deskripsi tentang suatu sesi multimedia [4], deskripsi tersebut antara lain mengenai :

- Nama dan tujuan *session*
- Waktu aktif *session* tersebut
- Tipe media yang akan digunakan
- *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 \*

#### **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)

## **2.4 Parameter-parameter QoS**

### **2.4.1 Delay**

*Delay* merupakan waktu yang dibutuhkan oleh sebuah paket data dari pengiriman oleh *transmitter* sampai diterima oleh *receiver*. Suatu aplikasi IPTV dibidang mempunyai QoS bagus bila *delay* yang terjadi kecil. Berikut ini adalah beberapa jenis *delay*, yaitu:

1. *Propagation delay* yaitu *delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima
2. *Serialization delay* yaitu *delay* yang terjadi pada saat proses perletakan bit ke dalam sirkuit

3. *Processing delay* yaitu *delay* yang terjadi saat proses *coding*, *compression*, *decompression* dan *decoding*
4. *Packetization delay* yaitu *delay* yang terjadi saat proses pemaketan *digital voice sample*
5. *Queuing delay* yaitu *delay* yang terjadi selama waktu tunggu paket sampai dilayani
6. *Jitter buffer* yaitu *delay* yang terjadi selama *buffer* untuk mengatasi *jitter*

#### 2.4.2 Packet Loss

*Packet loss* merupakan banyaknya paket data yang gagal mencapai tempat tujuan (sisi penerima) paket tersebut. *Packet loss* yang besar dapat terjadi ketika jaringan sedang sibuk atau terjadi *overload*. Pada IPTV, ketika video yang sedang dalam keadaan *streaming* dalam *variable rate mode* maka *packet loss* yang besar akan mengakibatkan penurunan kualitas IPTV.

## BAB 3

### PERANCANGAN DAN KONFIGURASI IPTV BERBASIS OPEN IMS

Pada perancangan IPTV ini digunakan *open source* yaitu OPEN IMS Core untuk mensimulasikan bagaimana cara kerja dari aplikasi IPTV. Untuk menjalankan IPTV ini, terlebih dahulu harus dikonfigurasi dan installasi beberapa komponen yakni: IMS Core, IMS client dan *application server*. IMS Core sendiri berfungsi sebagai inti dari IMS yang berperan dalam memproses semua permintaan dari *client* (pengguna) baik dari tahap register sampai permohonan untuk layanan aplikasi.

IMS client berfungsi sebagai *interface* dimana *client* dapat menggunakannya untuk melakukan proses registrasi, mengatur kualitas aplikasi yang diinginkan dan memanggil layanan aplikasi yang diinginkan. Sedangkan *application server* ini berfungsi dalam meneruskan permintaan dari *client* pada IMS Core dan mencocokkannya dengan *hash table* yang dimilikinya. Bila cocok maka *application server* akan mengirimkan layanan tersebut ke *client* sehingga *client* dapat menikmati layanan aplikasi yang diinginkannya.

#### 3.1 Open IMS Core

##### 3.1.1 Instalasi dan konfigurasi Open IMS Core

IMS Core yang digunakan dalam perancangan IPTV berbasis IMS ini adalah Open IMS Core. Sebelum melakukan konfigurasi dan instalasi IMS Core, anda bisa mengatur terlebih dahulu *server* tempat anda berada untuk mengunduh *file-file* yang nantinya dibutuhkan,

Berikut ini adalah langkah-langkah yang perlu dilakukan untuk menjalankan Open IMS Core:

1. *Update* repositori Ubuntu yang digunakan

**\$ sudo apt-get update**

2. Unduh dan instal *Subversion package* [5]

**\$ sudo apt-get install subversion**

3. Membuat *folder* OpenIMSCore

```
$ sudo mkdir /opt/OpenIMSCore/
```

4. Pengaturan kepemilikan pada *folder* OpenIMSCore

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

5. Membuat *folder* *ser\_ims* dan *FHoSS* di dalam *folder* OpenIMSCore:

```
$ cd /opt/OpenIMSCore
```

```
$ mkdir ser_ims
```

```
$ mkdir FHoSS
```

6. Melakukan *checkout* versi IMS Core terakhir dari *Berlios subversion servers* :

- a. Untuk CSCF

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

- b. Untuk FHoSS

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

7. Instal *package-package* yang dibutuhkan


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

8. Edit */etc/bind/named.conf* :

```
$ sudo nano /etc/bind/named.conf
```

Tambahkan baris berikut ini:

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



```

hartono@hartono-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/bind/named.conf Modified

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

include "etc/bind/named.conf.local"

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 3.1 Edit Named.conf

9. Salin file open-ims DNS ke *folder* bind

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

10. Edit /etc/resolv.conf [6]

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

Lalu edit seperti tampilan di bawah ini:



```

hartono@hartono-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/resolv.conf

### BEGIN INFO
#
# Modified by: NetworkManager
# Process: /usr/bin/NetworkManager
# Process_id: 5046
#
### END INFO

search open-ims.test
domain open-ims.test
nameserver 127.0.0.1

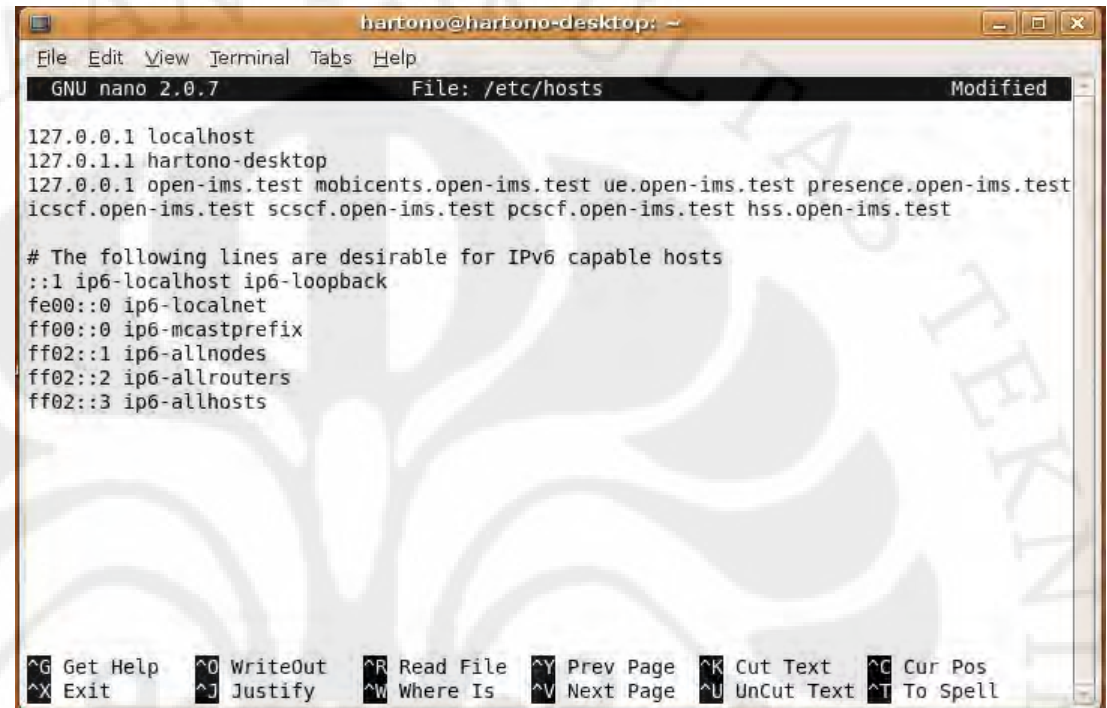
```

Gambar 3.2 Edit Resolv.conf

11. Edit /etc/hosts

**\$ sudo nano /etc/hosts**

Lalu edit seperti tampilan di bawah ini:



```

hartono@hartono-desktop: ~
File Edit View Terminal Tabs Help
GNU nano 2.0.7 File: /etc/hosts Modified
127.0.0.1 localhost
127.0.0.1 hartono-desktop
127.0.0.1 open-ims.test mobicents.open-ims.test ue.open-ims.test presence.open-ims.test
icscf.open-ims.test scscf.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

^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 Edit Hosts

12. *Restart* bind server

**\$ sudo /etc/init.d/bind9 restart**

13. Ping pcscf untuk mengecek langkah 8 sampai 12

**\$ ping pcscf.open-ims.test**

14. *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**

## 15. Compile source code

- Untuk CSCF:
 

```
$ cd /opt/OpenIMSCore/ser_ims
$ sudo make install-libs all
```
- Untuk FHoSS:
 

```
$ cd /opt/OpenIMSCore/FHoSS
$ ant compile deploy
```

## 16. Salin file konfigurasi ke folder OpenIMSCore

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

## 17. Jalankan OpenIMSCore

- masuk ke folder /opt/OpenIMSCore:
 

```
$ cd /opt/OpenIMSCore
```
- jalankan masing-masing CSCF di tab terminal yang berbeda:
 

```
$ ./pcscf.sh
```

```
hartono@hartono-laptop: /opt/OpenIMSCore
File Edit View Terminal Tabs Help
hartono@hart...  x hartono@hart...  x hartono@hart...  x hartono@hart...  x hartono@hart...  x
hartono@hartono-laptop:/opt/OpenIMSCore$ ./pcscf.sh
./pcscf.sh: line 4: setkey: command not found
./pcscf.sh: line 5: setkey: command not found
Listening on
      udp: 127.0.0.1 [127.0.0.1]:4060
      tcp: 127.0.0.1 [127.0.0.1]:4060
Aliases:
      tcp: localhost:4060
      udp: localhost:4060
      *: pcscf.open-ims.test:4060

0(2483) init_tcp: using epoll_lt as the io watch method (auto detected)
0(2483) Maxfwd module- initializing
0(2483) INFO:P-CSCF:mod_init: Initialization of module
0(2483) INFO:P-CSCF:mod_init: E-CSCF uri is sip:ecscf.open-ims.test:7060
0(2483) INFO:udp_init: SO_RCVBUF is initially 114688
0(2483) INFO:udp_init: SO_RCVBUF is finally 262142
1(2484) INFO:P-CSCF:mod_init: Initialization of module in child [1]
6(2489) INFO:P-CSCF:mod_init: Initialization of module in child [-1]
4(2487) INFO:P-CSCF:mod_init: Initialization of module in child [4]
0(2483) INFO:P-CSCF:mod_init: Initialization of module in child [0]
5(2488) INFO:P-CSCF:mod_init: Initialization of module in child [-1]
7(2490) INFO:P-CSCF:mod_init: Initialization of module in child [5]
```

Gambar 3.4 Running PCSCF



**\$ ./icscf.sh**

```

hartono@hartono-laptop: /opt/OpenIMSCore
File Edit View Terminal Tabs Help
hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x
hartono@hartono-laptop:/opt/OpenIMSCore$ ./icscf.sh
Listening on
  udp: 127.0.0.1 [127.0.0.1]:5060
  tcp: 127.0.0.1 [127.0.0.1]:5060
Aliases:
  tcp: localhost:5060
  udp: localhost:5060
  *: open-ims.test:*
  *: icscf.open-ims.test:*

0(2501) init_tcp: using epoll_lt as the io watch method (auto detected)
0(2501) Maxfwd module- initializing
0(2501) INFO:I-CSCF:mod_init: Initialization of module
0(2501) INF:I-CSCF:icscf_db_get_nds: Loaded 1 trusted domains
0(2501) INF:I-CSCF:icscf_db_get_capabilities: Loaded 2 capabilities for 1 S-CSC
Fs (0 invalid entries in db)
0(2501) Twofish encryption ready
0(2501) INFO:cdp:cdp_init(): CDiameterPeer initializing
0(2501) Diameter Peer Config:
0(2501)     FQDN      : icscf.open-ims.test
0(2501)     Realm     : open-ims.test
0(2501)     VendorID: 10415
0(2501)     ProdName: CDiameterPeer

```

Gambar 3.5 Running ICSCF

**\$ ./scscf.sh**

```

hartono@hartono-laptop: /opt/OpenIMSCore
File Edit View Terminal Tabs Help
hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x
hartono@hartono-laptop:/opt/OpenIMSCore$ ./scscf.sh
Listening on
  udp: 127.0.0.1 [127.0.0.1]:6060
  tcp: 127.0.0.1 [127.0.0.1]:6060
Aliases:
  tcp: localhost:6060
  udp: localhost:6060
  *: scscf.open-ims.test:6060

0(2527) init_tcp: using epoll_lt as the io watch method (auto detected)
0(2527) Maxfwd module- initializing
0(2527) INFO:S-CSCF:mod_init: Initialization of module
0(2527) INFO:ISC: - init
0(2527) INFO:cdp:cdp_init(): CDiameterPeer initializing
0(2527) Diameter Peer Config:
0(2527)     FQDN      : scscf.open-ims.test
0(2527)     Realm     : open-ims.test
0(2527)     VendorID: 10415
0(2527)     ProdName: CDiameterPeer
0(2527)     AcceptUn: [X]
0(2527)     DropUnkn: [X]
0(2527)     Tc       : 30
0(2527)     Workers  : 4

```

Gambar 3.6 Running SCSCF

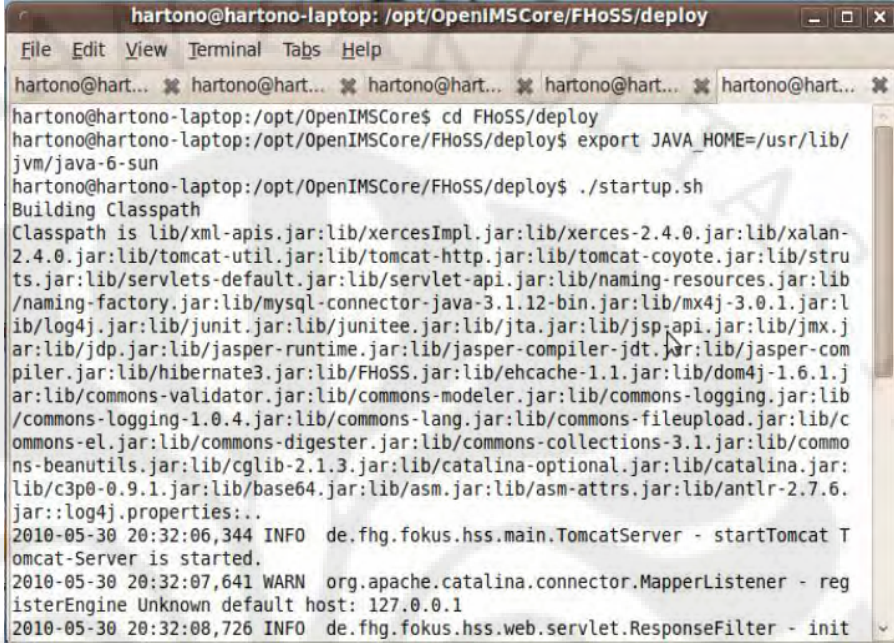
- masuk ke folder /opt/OpenIMSCore/FHoSS/deploy:

**\$ cd /opt/OpenIMSCore/FHoSS/deploy**

- Jalankan FHoSS:

```
$ export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

```
$ ./startup.sh
```



```
hartono@hartono-laptop: /opt/OpenIMSCore/FHoSS/deploy
File Edit View Terminal Tabs Help
hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x hartono@hart... x
hartono@hartono-laptop:/opt/OpenIMSCore$ cd FHoSS/deploy
hartono@hartono-laptop:/opt/OpenIMSCore/FHoSS/deploy$ export JAVA_HOME=/usr/lib/
jvm/java-6-sun
hartono@hartono-laptop:/opt/OpenIMSCore/FHoSS/deploy$ ./startup.sh
Building Classpath
Classpath is lib/xml-apis.jar:lib/xercesImpl.jar:lib/xerces-2.4.0.jar:lib/xalan-
2.4.0.jar:lib/tomcat-util.jar:lib/tomcat-http.jar:lib/tomcat-coyote.jar:lib/stru
ts.jar:lib/servlets-default.jar:lib/servlet-api.jar:lib/naming-resources.jar:lib
/naming-factory.jar:lib/mysql-connector-java-3.1.12-bin.jar:lib/mx4j-3.0.1.jar:l
ib/log4j.jar:lib/junit.jar:lib/junit4.jar:lib/jta.jar:lib/jsp-api.jar:lib/jmx.j
ar:lib/jdp.jar:lib/jasper-runtime.jar:lib/jasper-compiler-jdt.jar:lib/jasper-com
piler.jar:lib/hibernate3.jar:lib/FHoSS.jar:lib/ehcache-1.1.jar:lib/dom4j-1.6.1.j
ar:lib/commons-validator.jar:lib/commons-modeler.jar:lib/commons-logging.jar:lib
/commons-logging-1.0.4.jar:lib/commons-lang.jar:lib/commons-fileupload.jar:lib/c
ommons-el.jar:lib/commons-digester.jar:lib/commons-collections-3.1.jar:lib/commo
ns-beanutils.jar:lib/cglib-2.1.3.jar:lib/catalina-optional.jar:lib/catalina.jar:
lib/c3p0-0.9.1.jar:lib/base64.jar:lib/asm.jar:lib/asm-attrs.jar:lib/antlr-2.7.6.
jar::log4j.properties:..
2010-05-30 20:32:06,344 INFO de.fhg.fokus.hss.main.TomcatServer - startTomcat T
omcat-Server is started.
2010-05-30 20:32:07,641 WARN org.apache.catalina.connector.MapperListener - reg
isterEngine Unknown default host: 127.0.0.1
2010-05-30 20:32:08,726 INFO de.fhg.fokus.hss.web.servlet.ResponseFilter - init
```

Gambar 3.7 Running FHoSS

18. Elemen-elemen Open IMS Core beserta dengan portnya:

Tabel 2.1 Port Number Elemen OpenIMSCore

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

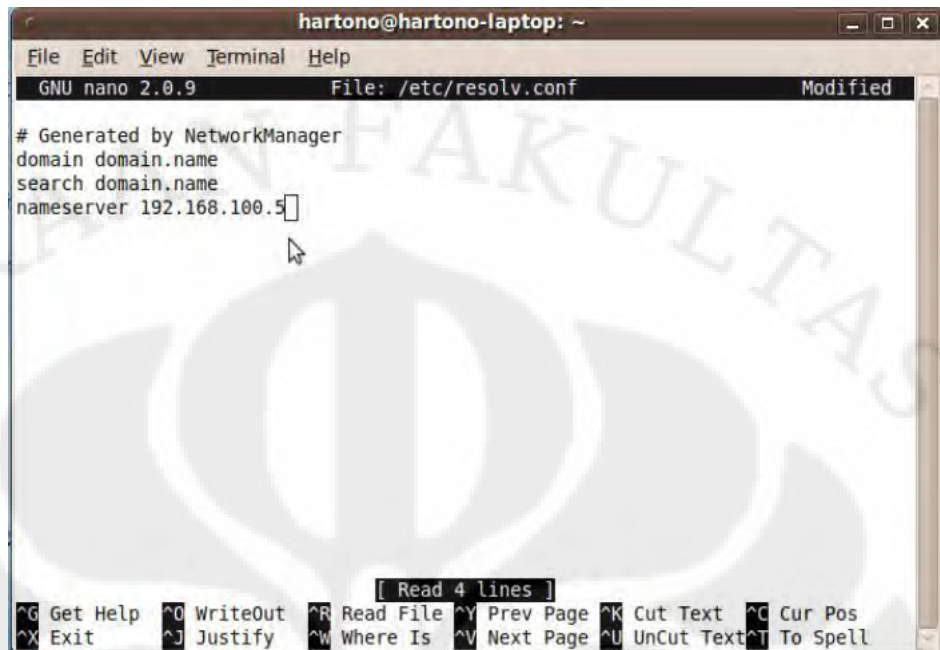
Element	Port Number
PCSCF	4060
ICSCF	5060
SCSCF	6060
DIAMETER	3868, 3869, 3870

### 3.1.2 Mengubah Open IMS Core ke Spesifik IP

Untuk mengubah Open IMS Core ke spesifik IP maka kita perlu mengganti semua IP 127.0.0.1 dengan IP yang kita inginkan dimana dalam skripsi ini digunakan IP 192.168.100.5. Berikut adalah langkah-langkahnya:

1. Jalankan perintah-perintah dibawah ini dan ubah IP 127.0.0.1 dengan 192.168.100.5

\$ sudo nano /etc/resolv.conf



```

hartono@hartono-laptop: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/resolv.conf Modified

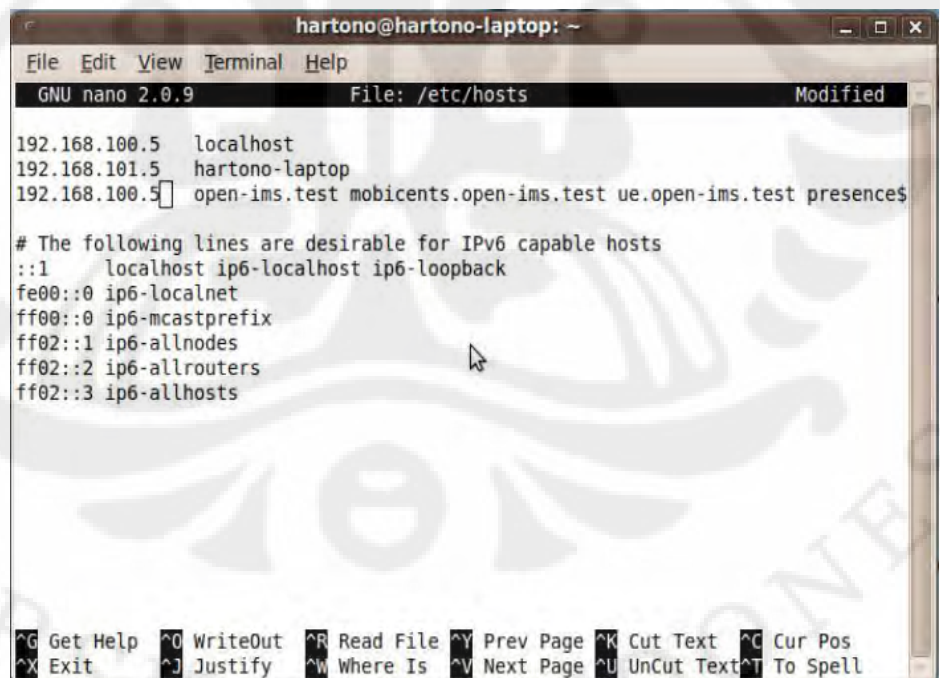
# Generated by NetworkManager
domain domain.name
search domain.name
nameserver 192.168.100.5

[ Read 4 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 3.8 Edit Resolv.conf ke Spesifik IP

\$ sudo nano /etc/hosts



```

hartono@hartono-laptop: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/hosts Modified

192.168.100.5 localhost
192.168.101.5 hartono-laptop
192.168.100.5 open-ims.test mobicents.open-ims.test ue.open-ims.test presences

# The following lines are desirable for IPv6 capable hosts
::1 localhost 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 Edit Hosts ke Spesifik IP

**\$ sudo nano /etc/bind/open-ims.dnszone**

```

hartono@hartono-laptop: ~
File Edit View Terminal Help
GNU nano 2.0.9 File: /etc/bind/open-ims.dnszone Modified
$ORIGIN open-ims.test.
$TTL 1W
@ 1D IN SOA 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.pscsf 1D SRV 0 0 4060 pcscf
_sip_udp.pscsf 1D SRV 0 0 4060 pcscf
_sip_tcp.pscsf 1D SRV 0 0 4060 pcscf
icscf 1D IN A 192.168.100.5
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Ctr Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 3.10 Edit Open-ims.dnszone ke Spesifik IP

2. Mengubah settingan dari *node-node* dengan perintah :

```

$ ./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 perintah di atas, maka masukkan 192.168.100.5 untuk IP dan open-ims.test untuk *realm*)

3. *Check dan restart* bind :

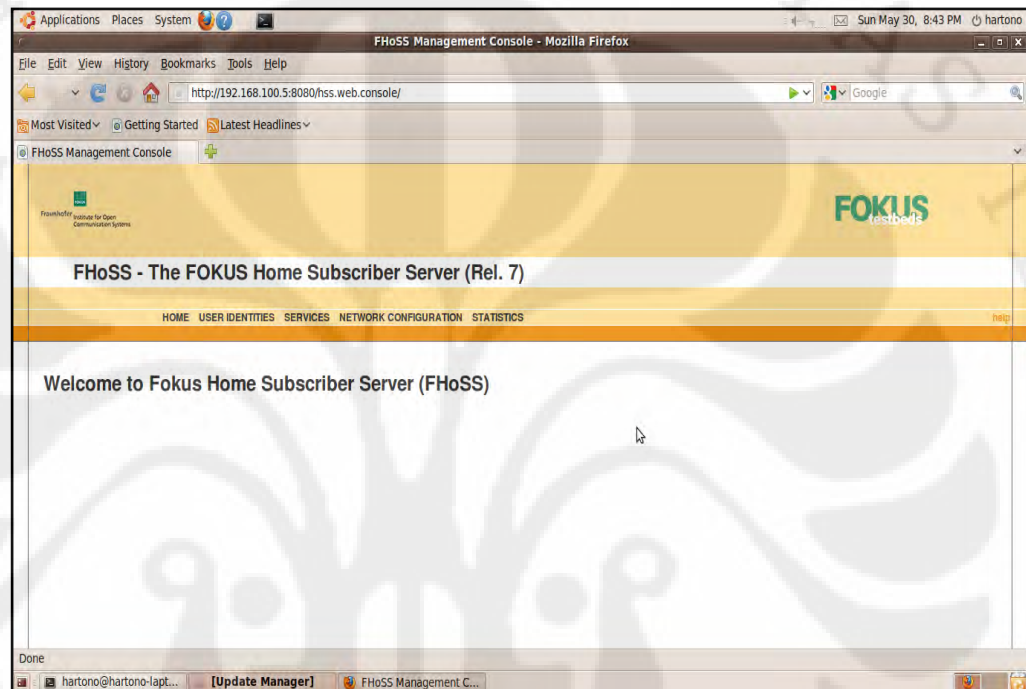
```

$ sudo /edit/init.d/bind9 restart
$ dig open-ims.test

```

#### 4. Membuka *web browser* OpenIMSCore

- Jalankan terlebih dahulu langkah ke-17
- Ketikkan 192.168.100.5:8080 pada *web address bar* di *web browser* kita
- Ketikkan hss\_Admin untuk *username* dan hss untuk *password*
- Bila berhasil, maka akan muncul tampilan berikut:



Gambar 3.11 FHoSS

### 3.2 Application Server IPTV

#### 3.2.1 Instalasi Application Server IPTV

- Unduh *file installer application server* IPTV dalam bentuk *debian packages* dari [https://developer.berlios.de/project/showfiles.php?group\\_id=7844](https://developer.berlios.de/project/showfiles.php?group_id=7844) [7]
- Instal *file* tersebut

#### 3.2.2 Konfigurasi Application Server IPTV

Untuk mengkonfigurasi *application server* IPTV maka kita perlu mengkonfigurasi FHoSS sehingga permintaan IPTV dapat dijalankan. Berikut adalah langkah-langkahnya:

1. Buat *application server* dengan nama *iptv* dan port: 7070

### Application Server -AS-

Application Server Details		Sh Interface - Permissions			
ID	-1	Permission for	UDR	PUR	SNR
Name*	iptv	Allowed Request	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Server Name*	sip:192.168.100.5:7070	Repository-Data	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diameter FQDN*	iptv.open-ims.test	IMPU	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Default Handling*	Session - Continued	IMS User State	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service Info		S-CSCF Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rep-Data Limit	1024	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-ISDN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Gambar 3.12 Application Server IPTV

2. Buat *trigger point* dengan nama IPTV\_trigger

### Trigger Point -TP-

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>ID</td> <td>2</td> </tr> <tr> <td>Name*</td> <td>IPTV_trigger</td> </tr> <tr> <td>Condition Type CNF*</td> <td>Conjunctive Normal Format</td> </tr> </table> <p style="font-size: small; text-align: center;">Mandatory fields were marked with ""</p> <p style="text-align: center;"> <input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Delete"/> </p>	ID	2	Name*	IPTV_trigger	Condition Type CNF*	Conjunctive Normal Format	<p>Attach IFC</p> <p>Select IFC... <input type="button" value="Attach"/></p> <p>List of attached IFCs</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID</th> <th>IFC Name</th> <th>Detach</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>iptv_filter</td> <td><input type="button" value="Detach"/></td> </tr> </tbody> </table>	ID	IFC Name	Detach	2	iptv_filter	<input type="button" value="Detach"/>																																																					
ID	2																																																																	
Name*	IPTV_trigger																																																																	
Condition Type CNF*	Conjunctive Normal Format																																																																	
ID	IFC Name	Detach																																																																
2	iptv_filter	<input type="button" value="Detach"/>																																																																
<p>Add SPTs to Trigger Point</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;">Not</td> <td style="width: 5%;"><input type="checkbox"/></td> <td style="width: 20%;">SIP Method</td> <td style="width: 40%;">INVITE</td> <td style="width: 10%;"><input type="button" value="Delete"/></td> </tr> <tr> <td colspan="5" style="text-align: center;">OR</td> </tr> <tr> <td colspan="5" style="text-align: center;">Request-URI <input type="button" value="Add"/></td> </tr> <tr> <td colspan="5" style="text-align: center;">AND</td> </tr> <tr> <td>Not</td> <td><input type="checkbox"/></td> <td>SIP Header</td> <td>To</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td colspan="2"></td> <td>SIP Header Content</td> <td>.*media.open-ims.test.*</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td colspan="5" style="text-align: center;">OR</td> </tr> <tr> <td>Not</td> <td><input type="checkbox"/></td> <td>SIP Header</td> <td>To</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td colspan="2"></td> <td>SIP Header Content</td> <td>.*iptv.open-ims.test.*</td> <td><input type="button" value="Delete"/></td> </tr> <tr> <td colspan="5" style="text-align: center;">OR</td> </tr> <tr> <td colspan="5" style="text-align: center;">Request-URI <input type="button" value="Add"/></td> </tr> <tr> <td colspan="5" style="text-align: center;">AND</td> </tr> <tr> <td colspan="5" style="text-align: center;">Request-URI <input type="button" value="Add"/></td> </tr> </table>		Not	<input type="checkbox"/>	SIP Method	INVITE	<input type="button" value="Delete"/>	OR					Request-URI <input type="button" value="Add"/>					AND					Not	<input type="checkbox"/>	SIP Header	To	<input type="button" value="Delete"/>			SIP Header Content	.*media.open-ims.test.*	<input type="button" value="Delete"/>	OR					Not	<input type="checkbox"/>	SIP Header	To	<input type="button" value="Delete"/>			SIP Header Content	.*iptv.open-ims.test.*	<input type="button" value="Delete"/>	OR					Request-URI <input type="button" value="Add"/>					AND					Request-URI <input type="button" value="Add"/>				
Not	<input type="checkbox"/>	SIP Method	INVITE	<input type="button" value="Delete"/>																																																														
OR																																																																		
Request-URI <input type="button" value="Add"/>																																																																		
AND																																																																		
Not	<input type="checkbox"/>	SIP Header	To	<input type="button" value="Delete"/>																																																														
		SIP Header Content	.*media.open-ims.test.*	<input type="button" value="Delete"/>																																																														
OR																																																																		
Not	<input type="checkbox"/>	SIP Header	To	<input type="button" value="Delete"/>																																																														
		SIP Header Content	.*iptv.open-ims.test.*	<input type="button" value="Delete"/>																																																														
OR																																																																		
Request-URI <input type="button" value="Add"/>																																																																		
AND																																																																		
Request-URI <input type="button" value="Add"/>																																																																		

Gambar 3.13 Trigger Point IPTV

3. Hubungkan *application server* dan *trigger point* dengan *Initial Filter Criteria* (IFC)

### Initial Filter Criteria -iFC-

ID	2
Name*	iptv_filter
Trigger Point	IPTV_trigger
Application Server*	iptv
Profile Part Indicator	Any

Mandatory fields were marked with "\*"

Gambar 3.14 Initial Filter Criteria IPTV

4. Masukkan IFC ke Shared iFC Set

### Shared iFC Sets -Sh-iFC-

ID-Set	1
Name*	default_shared_set

Mandatory fields were marked with "\*"

**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	<input type="button" value="Detach"/>
2	iptv_filter	2	<input type="button" value="Detach"/>

Gambar 3.15 Shared iFC Sets IPTV

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

**Service Profile -SP-**

ID	
Name*	default_sp
Core Network Service Auth	0

Mandatory fields were marked with \*\*\*

**Attach IFC**

Select IFC... Priority 0

**Attach Shared-IFC-Set**

Select Shared-IFC...

**List of attached IFCs**

ID	IFC Name	Priority	Detach
1	default_ifc	0	<input type="button" value="Detach"/>
2	iptv_filter	2	<input type="button" value="Detach"/>

**List of attached Shared-IFC-Sets**

ID-Set	Name	Detach
1	default_shared_set	<input type="button" value="Detach"/>

Gambar 3.16 Service Profile IPTV

### 3.3 IMS Client

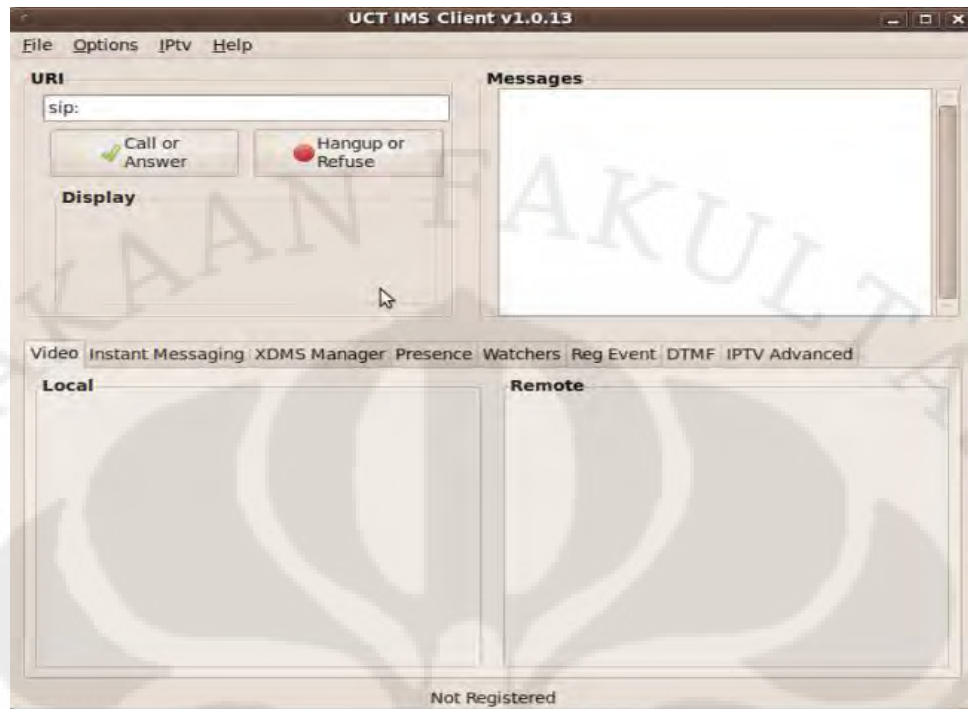
Untuk dapat memanggil aplikasi IPTV maka kita memerlukan perangkat yang dapat kita gunakan untuk melakukan registrasi dan memanggil aplikasi IPTV. Pada penelitian ini digunakan UCT IMS Client sebagai client.

#### 3.3.1 Instalasi dan konfigurasi IMS Client

- Unduh *file installer* UCT IMS Client dalam bentuk *debian packages* dari <http://uctimsclient.berlios.de/>
- Instal *file* *debian* tersebut
- Ketikkan perintah ini untuk memanggil UCT IMS Client:

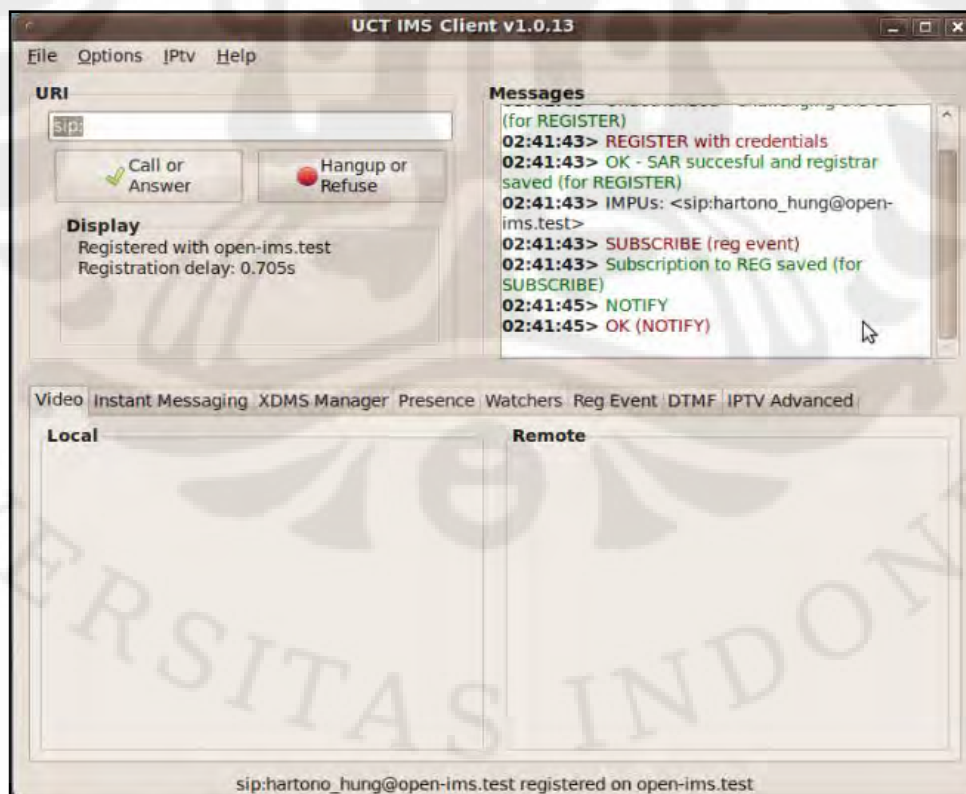
**\$ sudo uctimsclient**





Gambar 3.17 UCTIMSCIENT

- Register terlebih dahulu dengan *account client* (akan dijelaskan pada 3.3.2)



Gambar 3.18 Registrasi UCTIMSCIENT

- Pilih salah satu channel IPTV pada shortcut IPTV yang terdapat di kiri atas UCT IMS Client atau menyetikkan alamat URL channel IPTV yang diinginkan. Pada penelitian ini kita akan memanggil channel 1 maka kita tinggal menyetikkan alamat URL <sip:channel1@iptv.open-ims.test>.



Gambar 3.19 Tampilan IPTV

### 3.3.2 Membuat Account Client

Pada penelitian ini, akan dibuat *client* tambahan selain alice dan bob yang merupakan *client default* dari Open IMS. Oleh karena itu, kita perlu mengkonfigurasi halaman *web* FHoSS dengan langkah-langkah berikut:

1. Pada halaman *home* FHoSS, klik tab *USER IDENTITIES*
2. Klik *create* pada bagian *IMS Subscription*, lalu isi bagian *name* seperti tampilan di bawah ini dan kemudian tekan tombol *save*

**IMS Subscription -IMSU-**

ID	<input type="text" value="5"/>	<b>Create &amp; Bind new IMPI</b> <span style="color: green;">+</span>						
Name*	<input type="text" value="hartono_hung"/>	<b>Associate IMPI(s)</b> <input type="text" value="IMPI Identity"/> <input type="button" value="Add"/>						
Capabilities Set	<input type="text" value="none"/>							
Preferred S-CSCF	<input type="text" value="none"/>							
S-CSCF Name	<input type="text"/>							
Diameter Name	<input type="text"/>							
Mandatory fields were marked with "*"								
<input type="button" value="Save"/> <input type="button" value="Refresh"/> <input type="button" value="Delete"/>		<b>List of associated IMPIs</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">ID</th> <th style="width: 60%;">IMPI Identity</th> <th style="width: 20%;">Delete</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	ID	IMPI Identity	Delete			
ID	IMPI Identity	Delete						

Gambar 3.20 IMS Subscription

- Klik tanda + di bagian *Create & Bind new IMPI* pada tampilan di atas, lalu isi bagian *identity*, *secret key*, centang *tab all, default* seperti tampilan di bawah ini dan kemudian tekan tombol *save*

**Private User Identity -IMPI-**

ID	6
Identity*	hartono_hung@open-ims.test
Secret Key*	hartono_hung
<b>Authentication Schemes*</b>	
Digest-AKAv1 (3GPP)	<input type="checkbox"/>
Digest-AKAv2 (3GPP)	<input type="checkbox"/>
Digest-MD5 (FOKUS)	<input type="checkbox"/>
Digest (CableLabs)	<input type="checkbox"/>

**Associate an IMSU**

IMSU Identity:

**Associated IMSU**

ID	IMSU Identity	Delete
6	hartono_hung	<input type="button" value="Delete"/>

**Create & Bind new IMPU** +

**Associate IMPU(s)**

IMPU Identity:

Gambar 3.21 Bagian Atas Private User Identity –IMPI-

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

**List of associated IMPUs**

ID:	IMPU Identity:	Delete:

**Push Cx Operation**

Apply for:

**RTR Operation**

Apply for:

Gambar 3.22 Bagian Bawah Private User Identity –IMPI-

- Klik tanda + di bagian *Create & Bind new IMPU* pada tampilan di atas, lalu isi bagian *identity*, *service profile*, *display name* seperti tampilan di bawah ini dan kemudian tekan tombol *save*

### Public User Identity -IMPU-

ID	6
Identity*	sip:hartono_hung@open-ims.test
Barring	<input type="checkbox"/>
Service Profile*	default_sp
Implicit Set	6
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	hartono_hung
User-Status	NOT-REGISTERED

**Add Visited-Networks**

Select Visited-Network...

**List of Visited Networks**

ID	Identity	Delete

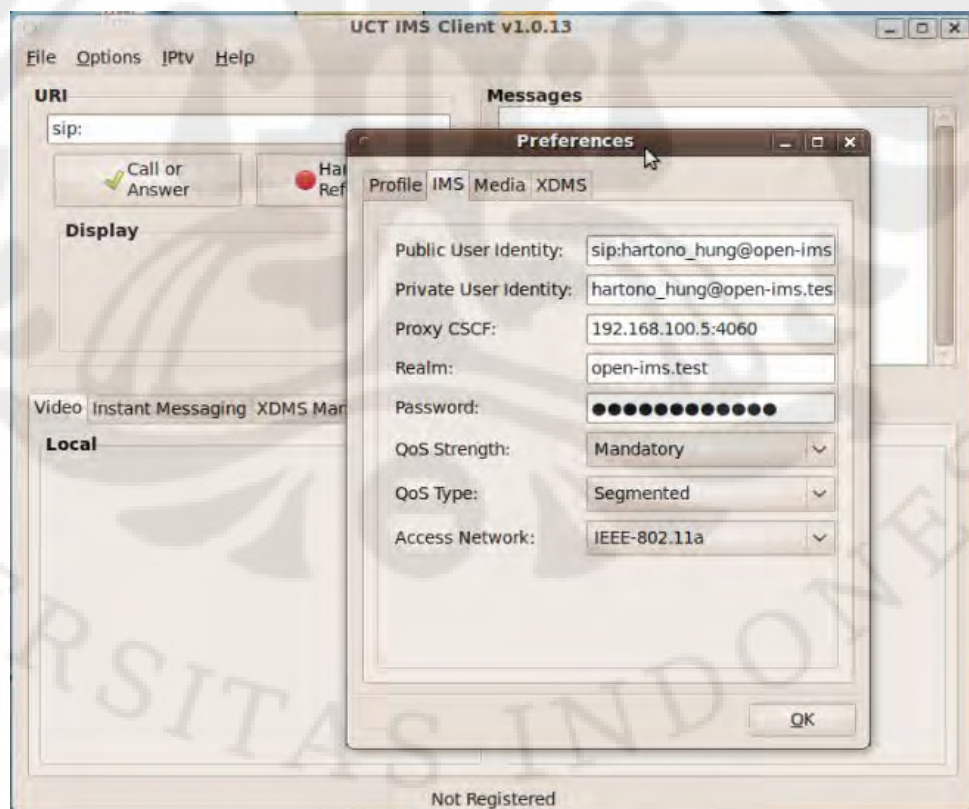
**Associate IMPI(s) to IMPU**

IMPI Identity

Warning: This IMPI will be associated with all the corresponding

Gambar 3.23 Public User Identity –IMPU-

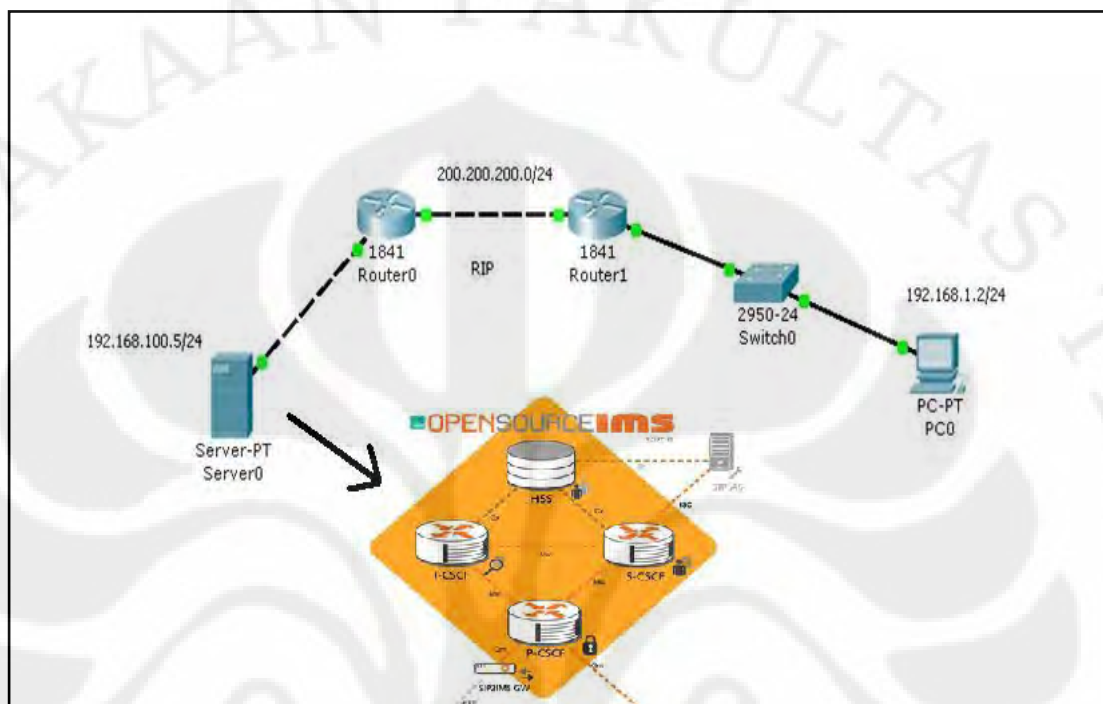
5. Buka kembali halaman pada tampilan di atas dan tambahkan open-ims.test pada bagian *Add Visited-Networks*
6. Setelah selesai, buka uctimsclient dan register dengan *account client* yang sudah dibuat.



Gambar 3.24 UCTIMSCLIENT Preferences

### 3.4 Topologi jaringan

Dalam penelitian ini, digunakan topologi jaringan seperti di bawah ini untuk mengemulasikan Open IMS, pemanggilan IPTV dan pengukuran data.



Gambar 3.25 Topologi Jaringan

Sesuai dengan Gambar 3.25, maka digunakan 2 buah *Personal Computer* (PC) sebagai router Vyatta, 1 buah Laptop sebagai *server* (OpenIMSCore), 1 buah laptop sebagai *client* dan 1 buah switch. Berikut ini adalah penjelasannya:

1. Antara komputer *client* dengan *switch* dihubungkan dengan kabel *straight through*
2. Antara *switch* dengan router1 dihubungkan dengan kabel *straight through*
3. Antara router1 dan router2 dihubungkan dengan kabel *crossover*
4. Antara server dengan router2 dihubungkan dengan kabel *crossover*

Pada sisi komputer *client* diberi *IP Address* 192.168.1.2, *netmask* 255.255.255.0 dan *gateway* 192.168.1.1. Pada sisi router1 di mana terdapat 2 *interface* Ethernet yaitu ethernet Eth0 yang terhubung ke router2 dan Eth1 yang terhubung ke komputer *client*. Ethernet Eth0 yang diberi *IP Address*

200.200.200.2, *netmask* 255.255.255.0 dan Eth1 yang diberi *IP Address* 192.168.1.1, *netmask* 255.255.255.0. Pada sisi router2 terdapat 2 *interface* Ethernet juga yaitu ethernet 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 sisi server diberi *IP Address* 192.168.100.5, *netmask* 255.255.255.0 dan *gateway* 192.168.100.1.

Skenario dari jaringan ini dimaksudkan untuk mengkondisikan proses registrasi *client*, *request* layanan IPTV oleh *client* dan *response* layanan IPTV oleh server seperti kondisi jaringan internet sebenarnya. Selain itu, kedua router vyatta yang digunakan juga diatur *bandwidth*-nya sedemikian rupa sehingga menggambarkan padatnya *traffic* pada internet. Berikut adalah pengaturan *bandwidth* pada kedua router:

1. Pada router1:
  - a. Jalankan perintah set **qos-policy traffic-shaper ABC** (nama yang ingin diberikan) **default bandwidth 123** (bandwidth yang diinginkan dalam angka, kbps)
  - b. Jalankan perintah set **interfaces ethernet Eth0 qos-policy out ABC**
  - c. **commit**
2. Pada router2:
  - a. Jalankan perintah set **qos-policy traffic-shaper ABC** (nama yang ingin diberikan) **default bandwidth 123** (bandwidth yang diinginkan dalam angka, kbps)
  - b. Jalankan perintah set **interfaces ethernet Eth0 qos-policy out ABC**
  - c. **commit**

Setelah jaringan terbangun, maka pada sisi *client* dijalankan program Wireshark untuk melihat proses registrasi sampai pemanggilan layanan IPTV, paket-paket data yang terjadi selama IPTV berjalan, proses deregistrasi dan menyimpan data-data tersebut sehingga nantinya bisa digunakan untuk menghitung *delay* dan *packet loss* yang terjadi.



## BAB 4

### HASIL PENGUKURAN DAN ANALISA PARAMETER QOS

Analisa parameter QoS pada implementasi IPTV menggunakan Open IMS Core ini, akan dilakukan berdasarkan data hasil pengukuran dari emulasi dan pengukuran data dengan menggunakan topologi jaringan seperti yang telah dijelaskan pada bab sebelumnya. Namun penulis akan terlebih dahulu menganalisa *message flow* dari registrasi *client*, *request* layanan IPTV sampai deregistrasi oleh *client*. Setelah itu, baru akan dibahas analisa parameter QoS beserta dengan data-data hasil pengukurannya.

#### 4.1 Message Flow

Untuk analisa *message flow*, *client* akan melakukan proses registrasi dengan menggunakan *software* IMS Client. *Public User Identity* yang digunakan untuk registrasi adalah [sip:hartono\\_hung@open-ims.test](mailto:sip:hartono_hung@open-ims.test). Setelah itu, *client* akan melakukan *request* layanan IPTV dengan mengetikkan alamat URL [sip:channell@iptv.open-ims.test](mailto:sip:channell@iptv.open-ims.test) dan pada akhirnya *client* akan melakukan pengakhiran layanan IPTV serta deregistrasi. Berikut ini adalah detail dari proses registrasi, *request* layanan IPTV dan pengakhirannya serta deregistrasi:

Tabel 4.1 Messages Flow

NO	Asal	Tujuan	Protocol	keterangan
1	<i>client</i>	IMS core	SIP	<i>Request: register sip:open-ims.test</i>
2	IMS core	<i>client</i>	SIP	Status: 401 <i>unauthorized</i>
3	<i>client</i>	IMS core	SIP	<i>Request: register sip:open-ims.test</i>
4	IMS core	<i>client</i>	SIP	Status: 200 ok – <i>SAR successful and registrar saved</i>
5	<i>client</i>	IMS core	SIP	<i>Reguest: SUBSCRIBE sip:hartono_hung@open-ims.test</i>
6	IMS core	<i>client</i>	SIP	Status: 200 <i>subscription to REG saved</i>
7	IMS core	<i>client</i>	SIP	<i>Request: Notify sip:hartono_hung@open-ims.test</i>



8	<i>client</i>	IMS core	SIP	Status: 200 ok
9	<i>client</i>	IMS core	SIP	<i>Request: Invite</i> <i>sip:channel@iptv.open-ims.test</i>
10	IMS core	<i>client</i>	SIP	Status: 100 <i>trying</i>
11	IMS core	<i>Application server</i>	SIP	<i>Request: Invite</i> <i>sip:channel1@iptv.open-ims.test</i>
12	IMS core	<i>client</i>	SIP	Status: 101 <i>dialog establishment</i>
13	IMS core	<i>client</i>	SIP	Status: 200 ok
14	<i>Application server</i>	<i>client</i>	RTP	<i>PT:dynamicRTP-Type-96</i>
15	<i>client</i>	<i>Application server</i>	SIP	<i>Request: bye sip:channel1@open-ims.test</i>
16	<i>Application server</i>	<i>client</i>	SIP	Status: 200 ok
17	<i>client</i>	IMS core	SIP	<i>Request: register sip:open-ims.test</i>
18	IMS core	<i>client</i>	SIP	Status: 401 <i>unauthorized</i>
19	<i>client</i>	IMS core	SIP	<i>Request: register sip:open-ims.test</i>
20	IMS core	<i>client</i>	SIP	Status: 200 <i>ok – SAR successful and registrar saved</i>
21	<i>client</i>	IMS core	SIP	Status: 200 <i>ok</i>

Sesuai dengan tabel di atas, penjelasan *message flow* yang terjadi antara *client*, *core*, dan *application server* sebagai berikut:

1. Pada langkah 1 sampai 8, *client* melakukan proses registrasi dengan *Public User Identity* [sip:hartono\\_hung@open-ims.test](mailto:sip:hartono_hung@open-ims.test) dan IMS core merespon bahwa terdapat *client* dengan identitas tersebut dan registrasi berhasil
2. Pada langkah 9, *client* melakukan *request* untuk menikmati layanan IPTV dengan *channel 1*
3. Pada langkah 10, IMS core memberi info kepada *client* kalau IMS core sedang mencoba menjalankan layanan IPTV channel 1

4. Pada langkah 11, *IMS core* meneruskan request *client* ke *application server*
5. Pada langkah 12, *core* memberitahu *client* bahwa *request* sudah bisa dijalankan
6. Pada langkah 13, *client* merespon bahwa *client* telah siap
7. Pada langkah 14, *application server* langsung mengirim paket-paket yang dibutuhkan untuk menonton channel 1 IPTV kepada *client*
8. Pada langkah 15, *client* melakukan proses pengakhiran layanan IPTV
9. Pada langkah 16, *application IMS core* mengirim info kepada *client* bahwa layanan IPTV sudah diakhiri
10. Pada langkah 17 sampai 21, *client* melakukan proses deregistrasi.

#### 4.2 Pengambilan dan pengolahan data

Pengambilan data dilakukan dengan merubah variasi nilai *bandwidth* dari jaringan yang digunakan yakni dengan mengatur besar *bandwidth* pada hubungan kedua router dengan nilai:

1. 64 kbps
2. 128 kbps
3. 256 kbps
4. 512 kbps
5. 768 kbps
6. 1024 kbps
7. 1280 kbps
8. 1536 kbps
9. 1792 kbps
10. 2048 kbps
11. 2112 kbps

Untuk setiap besar *bandwidth* dilakukan 3 kali pengambilan data dan nantinya akan didapatkan besar *delay* dan *packet loss* dengan menggunakan program Wireshark. Berikut ini adalah data hasil pengukuran dan pengolahannya :

1. *Bandwidth* = 64 kbps

Tabel 4.2 Packet Loss dan Delay Pada Bandwidth 64 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	92,37%	181,203
2	89,24%	175,649
3	93,21%	168,32
$\Sigma/n$ ( <i>Mean</i> )	91.61%	175,057

2. *Bandwidth* = 128 kbps

Tabel 4.3 Packet Loss dan Delay Pada Bandwidth 128 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	82,2%	104,958
2	78,64%	83,13
3	79,46%	106,972
$\Sigma/n$ ( <i>Mean</i> )	80,1%	98,353

3. *Bandwidth* = 256 kbps

Tabel 4.4 Packet Loss dan Delay Pada Bandwidth 256 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	65%	44,158
2	74,1%	50,572
3	75,7%	67,592
$\Sigma/n$ ( <i>Mean</i> )	72%	54,107

4. *Bandwidth* = 512 kbps

Tabel 4.5 Packet Loss dan Delay Pada Bandwidth 512 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	50,7%	28,348
2	50,3%	28,43
3	57,1%	27,366
$\Sigma/n$ ( <i>Mean</i> )	52,7%	28,048

5. *Bandwidth* = 768 kbps

Tabel 4.6 Packet Loss dan Delay Pada Bandwidth 768 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	38,6%	20,041
2	44%	21,832
3	40%	21,531
$\Sigma/n$ ( <i>Mean</i> )	40,87%	21,135

6. *Bandwidth* = 1024 kbps

Tabel 4.7 Packet Loss dan Delay Pada Bandwidth 1024 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	29,6%	17,578
2	28,8%	17,636
3	32,2%	20,452
$\Sigma/n$ ( <i>Mean</i> )	30,2%	18,555

7. *Bandwidth* = 1280

Tabel 4.8 Packet Loss dan Delay Pada Bandwidth 1280 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	24,6%	16,245
2	21,1%	16,932
3	24,4%	15,778
$\Sigma/n$ ( <i>Mean</i> )	23,37%	16,318

8. *Bandwidth* = 1536 kbps

Tabel 4.9 Packet Loss dan Delay Pada Bandwidth 1536 kbps

Data	<i>Packet Loss</i> (%)	<i>Delay</i> (ms)
1	10,3%	15,384
2	14,2%	15,458
3	11,32%	14,518
$\Sigma/n$ ( <i>Mean</i> )	12,25%	15,12

9. *Bandwidth* = 1792 kbps

Tabel 4.10 Packet Loss dan Delay Pada Bandwidth 1792 kbps

Data	<i>Packet Loss (%)</i>	<i>Delay (ms)</i>
1	7,43%	14,647
2	6,57%	15,325
3	8,81%	12,917
$\Sigma/n$ ( <i>Mean</i> )	7%	14,296

10. *Bandwidth* = 2048 kbps

Tabel 4.11 Packet Loss dan Delay Pada Bandwidth 2048 kbps

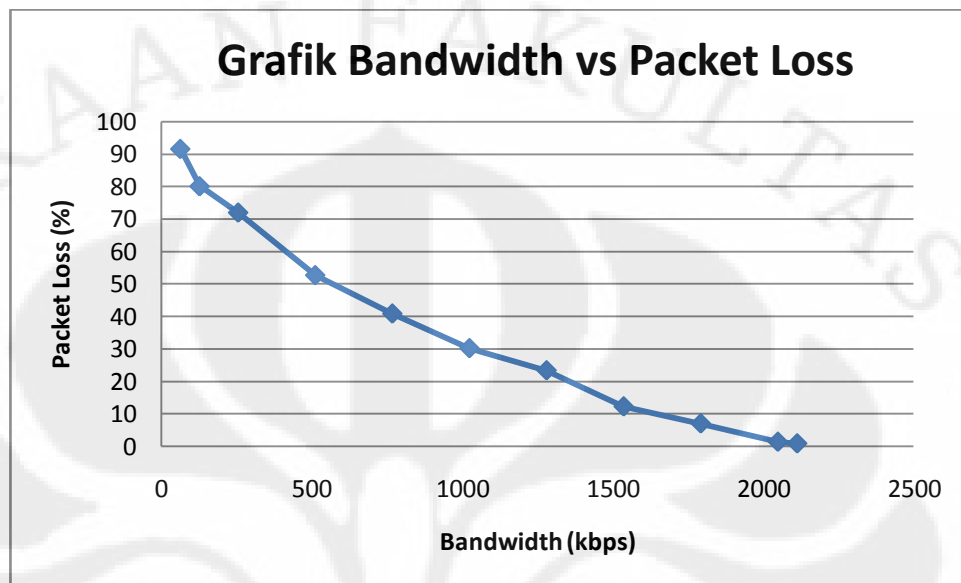
Data	<i>Packet Loss (%)</i>	<i>Delay (ms)</i>
1	1,34%	13,677
2	1,43%	7,774
3	1,28%	16,296
$\Sigma/n$ ( <i>Mean</i> )	1,35%	12,774

11. *Bandwidth* = 2112 kbps

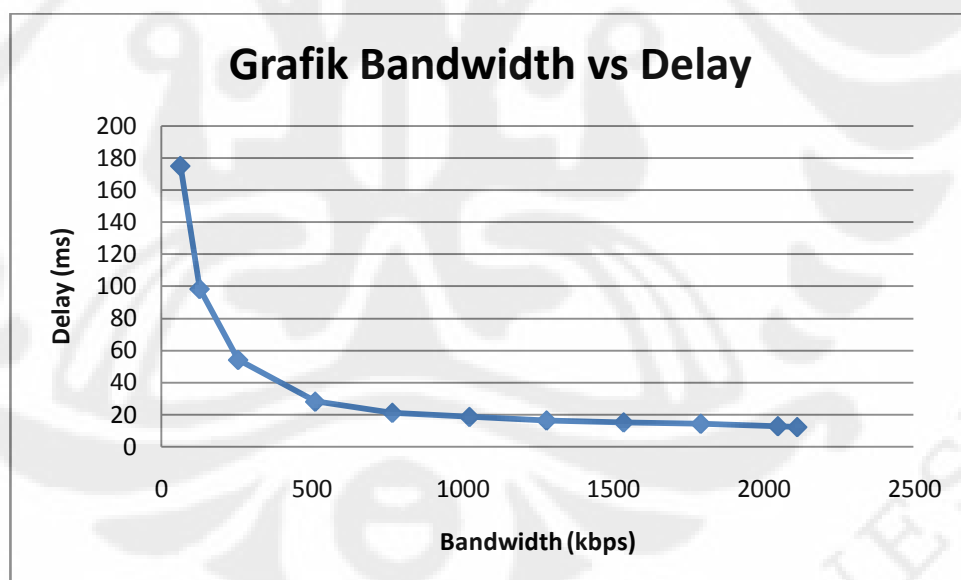
Tabel 4.12 Packet Loss dan Delay Pada Bandwidth 2112 kbps

Data	<i>Packet Loss (%)</i>	<i>Delay (ms)</i>
1	0,86%	13,451
2	0,93%	7,253
3	0,75%	15,984
$\Sigma/n$ ( <i>Mean</i> )	0,85%	12,229

Berikut adalah grafik perbandingan antara bandwidth dengan packet loss dan delay:



Gambar 4.1 Grafik Perbandingan Bandwidth dengan Packet Loss



Gambar 4.2 Grafik Perbandingan Bandwidth dengan Delay

### 4.3 Analisa parameter QoS

Sesuai dengan data-data pengukuran yang telah didapatkan sebelumnya, kita telah mendapatkan dua parameter QoS yang terdiri dari: *packet loss* dan

*delay*. Pada pengaturan bandwidth 64 kbps didapatkan nilai packet loss dan delay yang besar yakni 91,16% dan 175,057 ms. Packet loss ini terjadi karena kecilnya bandwidth yang ada di jaringan sehingga sebagian besar data akan ditinggalkan (drop) untuk menjaga kestabilan jaringan, sedangkan delay terjadi karena rentang waktu yang dibutuhkan dalam pemaketan paket data untuk melalui jaringan yang digunakan untuk pengukuran data ini.

Dengan seiring ditingkatkannya besar bandwidth maka besar packet loss dan delay ini menjadi lebih kecil sehingga dengan menaikkan nilai bandwidth secara signifikan maka didapatkan bandwidth optimum untuk implementasi aplikasi IPTV yaitu 2112 kbps. Pada bandwidth 2112 kbps ini, didapatkan nilai packet loss = 0,85 % dan delay = 12,229 ms sehingga dapat dikatakan bahwa aplikasi IPTV sudah layak diimplementasikan pada besar bandwidth ini karena sesuai dengan standar ITU-T, nilai *packet loss* maksimum adalah 1%, sedangkan untuk *delay* adalah 10 s [4].

## BAB 5

### KESIMPULAN

Pada skripsi ini yang berjudul “Analisa Implementasi IPTV Dengan Menggunakan Open IMS Core”, dapat disimpulkan beberapa hal dari pengujian yang telah dilakukan yaitu:

1. Jaringan IMS yang telah dibangun berjalan dengan baik dan sesuai dengan konsep kerjanya
2. Pemakaian dua router di antara komputer *server* dan *client* dimaksudkan untuk mengkondisikan jaringan yang dipakai untuk implementasi IPTV ini seperti jaringan internet yang sebenarnya. Di mana bila kita ingin mengakses layanan IPTV maka harus melewati router dan *traffic* jaringan.
3. Dengan menaikkan besar nilai bandwidth, maka nilai packet loss dan delay akan berkurang
4. *Bandwidth* optimum dalam implementasi IPTV menggunakan Open IMS Core ini adalah 2112 kbps, di mana nilai *packet loss* = 0,85 % dan *delay* = 12,229 ms.



## DAFTAR REFERENSI

- [1] Juliet Bates, et al., “Converged Multimedia Network” (hal 71), John Wiley & Sons, UK, 2006.
- [2] Dragos, V, Weik, P, & Magedanz, T, “Design and Implementation of an Open IMS Core” (hal 286-287), Springer Berlin / Heidelberg, Berlin, Germany, 2005.
- [3] Hens, F.J., & Caballero, J.M., “TRIPLE PLAY Building the Converged Network for IP, VoIP and IPTV” (hal 52), Trend Communications Ltd., UK, 2008
- [4] Hens, F.J., & Caballero, J.M., “TRIPLE PLAY Building the Converged Network for IP, VoIP and IPTV” (hal 121-123), Trend Communications Ltd., UK, 2008
- [4] ITU-T (2001, November). End-user Multimedia QoS Categories (ITU-T Rec. G.1010).
- [5] How to Install the Open IMS Core on Ubuntu. Diakses 10 april 2010 dari [uctimsclient.berlios.de](http://uctimsclient.berlios.de)  
[http://uctimsclient.berlios.de/openimscore\\_on\\_ubuntu\\_howto.html](http://uctimsclient.berlios.de/openimscore_on_ubuntu_howto.html)
- [6] OpenIMScore Installation on Ubuntu 9.04. Diakses 15 april 2010 dari [raziyabhayani.blogspot.com](http://raziyabhayani.blogspot.com)  
<http://raziyabhayani.blogspot.com/2009/07/openimscore-installation-on-ubuntu-904.html>
- [7] How to Run the UCT IPTv Streaming Server. Diakses 20 mei 2010 dari [uctimsclient.berlios.de](http://uctimsclient.berlios.de)  
[http://uctimsclient.berlios.de/uctiptv\\_howto.html](http://uctimsclient.berlios.de/uctiptv_howto.html)