UNIVERSITAS INDONESIA

PERANCANGAN DAN IMPLEMENTASI SISTEM CHARGING UNTUK LAYANAN IPTV (VOD) PADA IMS

SKRIPSI

CHANDRA GUNAWAN

0706267585

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

TEKNIK ELEKTRO

DEPOK

JUNI 2011

UNIVERSITAS INDONESIA

PERANCANGAN DAN IMPLEMENTASI SISTEM CHARGING UNTUK LAYANAN IPTV (VOD) PADA IMS

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar Sarjana Teknik

CHANDRA GUNAWAN
0706267585

FAKULTAS TEKNIK UNIVERSITAS INDONESIA
TEKNIK ELEKTRO
TELEKOMUNIKASI
DEPOK
JUNI 2011

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama	**: Chandra Gunawan**
NPM	**: 0706267585**
Tanda Tangan	**:** 
Tanggal	**: 13 JUNI 2011**

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh	:
Nama	: Chandra Gunawan
NPM	: 0706267585
Program Studi	: Teknik Elektro
Judul Skripsi	: Perancangan dan Implementasi Sistem Charging untuk Layanan IPTV (VoD) pada IMS

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. Anak Agung Putri Ratna M.Eng (.....)

Penguji : Ir. Endang Sriningsih MT, Si (.....)

Ditetapkan di : Depok
Tanggal : 27 Juni 2011

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia.

Atas proses bimbingan yang telah dijalani dan proses penyusunan dari skripsi ini, penulis ingin mengucapkan terima kasih kepada:

1. Ir. Djamhari Sirat M. Sc., Ph.D, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan skripsi ini;
2. Prof. Dr. Ir. Bagio Budiardjo M.Sc. dan Muhammad Salman ST., MIT, selaku pembimbing DTE IMS Research Group;
3. Pak Yulianus, yang telah membantu dalam pembahasan tentang topik *OpenIMSCore*;
4. Pak Randi, yang telah membantu dalam pembahasan tentang IMS;
5. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
6. Rekan-rekan satu bimbingan: Ardy Thiotrisno, Rosa, Faisal Jamil, dan Krisna Juanta, yang telah membantu dan memotivasi dalam membangun jaringan implementasi *OpenIMSCore*;
7. Asisten laboratorium jaringan, yang telah membantu penulis dalam proses instalasi *OpenIMSCore*;
8. Sahabat-sahabat penulis: Rizky Prasetya, Muhammad Rifki N., M. Wahyu Ashari, Abdullah Umar, Rizky Agung Tri Atmaja, Rhyando Anggoro Adi, Victor, dan lain-lain, yang telah membantu dan memberikan semangat penulis dalam menyelesaikan skripsi ini.
9. Seluruh keluarga besar Civitas Akademika Fakultas Teknik Universitas Indonesia khususnya karyawan sekretariat Departemen Teknik Elektro yang telah banyak memberikan bantuan dalam urusan administrasi.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

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

Depok, 13 Juni 2011

Penulis

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama	: Chandra Gunawan
NPM	: 0706267585
Program Studi	: Teknik Elektro
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis Karya	: Skripsi

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

Perancangan dan Implementasi Sistem *Charging* untuk Layanan IPTV (VoD) pada IMS

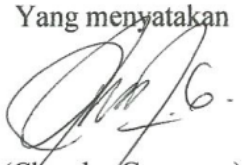
Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, danmempublikasikan tugas akhir saya selama tetap mencatumkan nama saya sebagai penulis/pencipta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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 13 Juni 2011

Yang menyatakan



(Chandra Gunawan)

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

ABSTRAK

Nama : Chandra Gunawan
Program Studi : Teknik Elektro
Judul : Perancangan dan Implementasi Sistem Charging untuk Layanan IPTV (VoD) pada IMS

Skripsi ini membahas tentang implementasi sistem *charging* untuk layanan IPTV (VoD) pada IMS. IMS merupakan salah satu *platform* berbasis IP yang mendukung adanya konvergensi telekomunikasi. Salah satu layanan IMS yang akan berkembang adalah IPTV, dan salah satu aplikasinya adalah *Video on Demand (VoD)*. Dengan adanya VoD, pengguna layanan dapat memilih dan menonton video sesuai dengan keinginannya. Selain itu, pengguna layanan juga bisa kapan saja menonton video pilihan mereka sendiri.

Seperti yang distandarkan oleh 3GPP, sistem *charging* pada IMS terdiri dari beberapa elemen utama, yaitu *Charging Trigger Function (CTF), Charging Data Function (CDF),* dan *Online Charging Function (OCF).*

Pada implementasi sistem *charging* untuk layanan IPTV (VoD) ini, digunakan *UCT IMS Charging System* yang telah dikembangkan oleh University of Cape Town, *application server, media server, OpenIMSCore,* dan *software interface* perhitungan biaya. Sistem *charging* ini dapat digunakan untuk *online charging* dan *offline charging.* Setelah memperoleh data dari *UCT IMS Charging System,* data tersebut kemudian akan masuk secara otomatis ke dalam *software interface* perhitungan biaya untuk dilakukan penghitungan total biaya yang dikenakan kepada *client* atas penggunaan layanan IPTV (VoD).

Kata kunci:
konvergensi, IMS, IPTV (VoD), *charging, UCT IMS Charging System, application server, media server, OpenMSCore, software interface* perhitungan biaya

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

ABSTRACT

Name : Chandra Gunawan
Study Program : Teknik Elektro
Title : Design and Implementation of Charging System for IPTV (VoD) Service in IMS

The focus of this thesis is the implementation of charging system for IPTV (VoD) service in IMS. IMS is one of the IP based platform that supports the convergence of telecommunications. One of the IMS services that will be evolve is IPTV, and one of its applications is Video on Demand (VoD). With VoD, users can select and watch the videos as they wish. In addition, users can also watch their selected videos anytime they want.

As standardized by 3GPP, IMS charging involves the following main entities: a Charging Trigger Function (CTF), a Charging Data Function (CDF), and Online Charging Function (OCF).

In this charging system, UCT IMS Charging System which has been developed by University of Cape Town is used. Application server, media server, OpenIMSCore, and interface software costing is used as well. Charging system for IPTV (VoD) service can be used for online charging and offline charging. After obtaining the data from the UCT IMS Charging System, the data will be entered automatically into interface software costing for a head count of the total fees charged to client for the use of IPTV (VoD) service.

Key words:
convergence, IMS, IPTV (VoD), charging, UCT IMS Charging System, application server, media server, OpenIMSCore, interface software costing

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

DAFTAR ISI

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

DAFTAR TABEL

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

DAFTAR GAMBAR

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

DAFTAR SINGKATAN

3GPP	*3rd Generation Partnership Project*
ACA	*Accounting Answer*
ACR	*Accounting Request*
AS	*Application Server*
AVP	*Attribute Value Pair*
CAPEX	*Capital Expenditure*
CCA	*Credit Control Answer*
CCR	*Credit Control Request*
CDF	*Charging Data Function*
CGF	*Charging Gateway Function*
CTF	*Charging Trigger Function*
EU	*End User*
GPRS	*General packet Radio Service*
HLR	*Home Location Register*
HSS	*Home Subscriber Server*
HTTP	*Hypertext Transfer Protocol*
I-CSCF	*Interrogating Call State Control Function*
IFC	*Initial Filter Criteria*
IM	*Instant Messaging*
IMS	*IP Multimedia Subsystem*
IP	*Internet Protocol*
IPTV	*Internet Protocol Television*
MOD	*Manufacturing on Demand*
MRCF	*Media Resource Control Function*
MRF	*Media Resource Function*
NGN	*Next Generation Network*
NVOD	*Near Video On Demand*
OCF	*Online Charging Function*
OCS	*Online Charging System*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

OPEX	*Operational Expenditure*
P-CSCF	*Proxy Call State Control Function*
PLMN	*Public Land Mobile Network*
PSTN	*Public Switched Telephone Network*
PVOD	*Push Video On Demand*
RSVP	*Resource Reservation Protocol*
RTCP	*Real Time Control Protocol*
RTP	*Real Time Transport Protocol*
RTSP	*Real Time Streaming Protocol*
S-CSCF	*Serving Call State Control Function*
SDP	*Session Description Protocol*
SIP	*Session Initiation Protocol*
SMRP	*Simple Multicast Routing Protocol*
UCT	*University of Cape Town*
URIs	*Uniform Resorce Indicators*
VoD	*Video on Demand*
VoIP	*Voice over IP*
WLAN	*Wireless Local Area Network*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Perkembangan dunia telekomunikasi saat ini telah mengalami perkembangan yang sangat pesat. Persaingan antar operator selular maupun vendor pun semakin ketat dalam mengadakan pelayanan-pelayanan kepada masyarakat yang berhubungan dengan dunia telekomunikasi. Oleh karena itu, ‘‘perang tarif’’ pun terjadi. Setiap operator selular berlomba-lomba untuk memberikan tarif yang semurah mungkin untuk menarik pelanggan.

Pada saat ini, sebagian infrastruktur telekomunikasi di Indonesia masih menggunakan metode *circuit switching network* untuk menghubungkan pelanggan telekomunikasi yang satu dengan yang lainnya. Metode ini menghubungkan satu kanal sambungan telepon dengan kanal yang lain dan menjamin kehandalan dalam berkomunikasi satu sama lain. Akan tetapi, yang menjadi kekurangan dari metode ini adalah memerlukan biaya yang tinggi baik untuk biaya operasional maupun biaya infrastruktur. Selain itu, untuk mengembangkan metode *circuit switching* ini membutuhkan waktu yang relatif lama karena tingkat kesulitan yang tinggi dalam proses pengembangannya. Oleh karena itu, diperlukan suatu metode baru yang dapat menanggulangi kekurangan-kekurangan yang ada pada metode *circuit switching* tersebut. Kemudian muncullah suatu metode baru yang dikenal dengan *packet switching.* Pada metode ini, suatu informasi atau data dibagi menjadi paket-paket yang kemudian dikirimkan ke alamat yang dituju.

Dengan munculnya metode *packet switching network,* seluruh jaringan *transport* dan akses akan berbasis IP. Salah satu layanan berbasis IP yang sedang dikembangkan saat ini adalah IPTV yang salah satu aplikasinya adalah *Video on Demand* (VoD). Jenis layanan ini

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

memungkinkan masyarakat dapat mengakses seluruh acara TV di seluruh dunia dengan mudahnya dan dapat menentukan acara TV atau video yang ingin mereka tonton (VoD). Dengan demikian, masyarakat menjadi lebih mudah untuk menonton acara TV atau video dan tidak perlu mengeluarkan biaya tambahan untuk membeli pemancar TV untuk bisa mengaksesnya.

Salah satu *platform* yang menyediakan layanan IPTV (VoD) adalah IMS (*IP Multimedia Subsystem*), dimana pada *platform* ini berbagai jenis layanan dikemas menjadi satu *platform* dan dijual kepada pelanggan telekomunikasi secara terintegrasi. Dengan demikian, diperlukan suatu sistem *charging* yang dapat diterapkan pada *platform* tersebut untuk memfasilitasi pengumpulan pendapatan atas penggunaan sumber daya (layanan) [1]. Oleh karena itu, pada skripsi ini akan dibahas suatu implementasi sistem *charging* untuk layanan IPTV (VoD).

1.2 Perumusan Masalah

Masalah yang dibahas pada skripsi ini berkisar pada implementasi sistem *charging* untuk layanan IPTV (VoD), secara rinci masalah yang akan dibahas adalah:

1. Elemen-elemen apa saja yang dibutuhkan?
2. Bagaimana konfigurasi elemen-elemen tersebut?
3. Bagaimana konsep perhitungan *charging*-nya?

1.3 Tujuan

Tujuan dari penulisan skripsi ini adalah:

1. Mengetahui elemen-elemen apa saja yang terlibat pada perancangan sistem *charging* untuk layanan IPTV (VoD).
2. Memahami *message flow* penggunaan layanan IPTV (VoD) pada IMS.
3. Meng-*explore* sistem *charging* untuk layanan IPTV (VoD) pada *Open Source IMS* dan merancang sistem perhitungan biayanya.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

1.4 Pembatasan Masalah

Pada skripsi ini, masalah dibatasi pada implementasi sistem *charging* untuk layanan IPTV (VoD) pada *Open Source IMS*. Untuk melakukan hal tersebut, diperlukan *IMS Core*, *application* server, *media server,* dan *software interface* untuk melakukan perhitungan biayanya.

1.5 Metodologi Penulisan

Metodologi penulisan yang digunakan pada penulisan skripsi ini adalah studi pustaka dan uji coba dengan menggunakan *Open Source IMS* dan *software interface* perhitungan biaya, serta melakukan analisis dengan menggunakan wireshark.

1.6 Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi beberapa bab, yaitu:

BAB 1 PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang, perumusan masalah, tujuan, pembatasan masalah, metodologi penulisan, dan sistematika penulisan.

BAB 2 IMS SEBAGAI KONVERGENSI TELEKOMUNIKASI

Pada bab ini akan dijelaskan mengenai konvergensi telekomunikasi, *IP Multimedia Subsystem (IMS), Video on Demand (VoD),* protokol-protokol pada layanan multimedia, dan Sistem *charging* pada IMS.

BAB 3 PERANCANGAN SISTEM *CHARGING* UNTUK LAYANAN IPTV (VOD)

Bab ini menjelaskan tentang skenario uji coba, *OpenIMSCore*, *application server*, *media server*, pengetesan IPTV (VoD), dan *UCT IMS Charging System*.

BAB 4 HASIL UJI COBA DAN ANALISIS

Bab ini memaparkan tentang uji coba dan hasilnya, serta analisis terhadap hasil uji coba tersebut.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 5 KESIMPULAN

Bab ini berisi kesimpulan yang didapat berdasarkan pembahasan-pembahasan yang ada di dalam skripsi ini.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 2
IMS SEBAGAI KONVERGENSI TELEKOMUNIKASI

2.1 Konvergensi Telekomunikasi

Konvergensi merupakan suatu integrasi dari beberapa *platform* layanan dan teknologi yang berbeda menjadi satu *platform* yang sama [2]. Dengan adanya konvergensi ini, layanan-layanan telekomunikasi dari *platform-platform* yang berbeda bisa diakses dari satu *platform*. Salah satu *platform* yang menyediakan konvergensi ini adalah IMS. Adapun ciri-ciri dari jaringan konvergensi adalah:

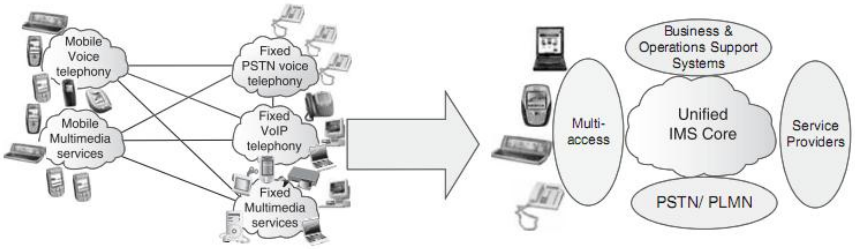
a. Jaringannya berbasis IP atau *packet switching network.*
b. Aplikasi/layanan yang ada terpisah dengan jaringan *transport*, terletak pada *device* masing-masing layanan yang akan digunakan.
c. Setiap orang bisa terhubung dengan jaringan ini karena jaringan ini bersifat terbuka.
d. Jaringan *broadband.*
e. Jaringan ini bersifat *ubiquitous,* bisa diakses dimana dan kapan saja.

Konvergensi dapat dilihat dari tiga sudut pandang [3], yaitu:

1) Konvergensi Jaringan

Konvergensi jaringan memudahkan EU (*end user*) dalam mengakses berbagai macam layanan dan meminimalisir hambatan-hambatan serta kompleksitas yang ada pada jaringan saat ini. Jika dilihat dari sudut pandang *operator,* tujuan dari konvergensi jaringan ini adalah melakukan perubahan dari jaringan-jaringan PSTN, PLMN, jaringan utama, dan jaringan IP yang terpisah menjadi satu jaringan yang mendukung teknologi-teknologi akses tersebut. Perubahan ini akan menjadi kunci bagi sebuah *operator* untuk mengurangi OPEX dan CAPEX, dan meningkatkan daya saing dan profitabilitas.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 2.1 Konvergensi Jaringan [3]

2) Perangkat Konvergensi

Untuk saat ini, suatu perangkat hanya dapat digunakan untuk mengakses suatu layanan pada jaringan yang sama. Jika konsumen ingin mengakses layanan lain yang membutuhkan perpindahan jaringan akses, maka ia harus mengganti perangkat yang digunakannya tersebut. Hal ini dikarenakan adanya suatu *gateway* untuk mengakses suatu layanan yang berbeda jaringan. Oleh karena itulah, dibutuhkan suatu perangkat yang dapat digunakan untuk mengakses layanan-layanan yang berbeda jaringan tersebut. Dengan adanya konvergensi, konsumen dapat mengakses layanan-layanan yang ada tanpa harus mengganti perangkat dan perangkat yang digunakan tersebut dikenal dengan nama perangkat konvergensi.

3) Layanan Konvergensi

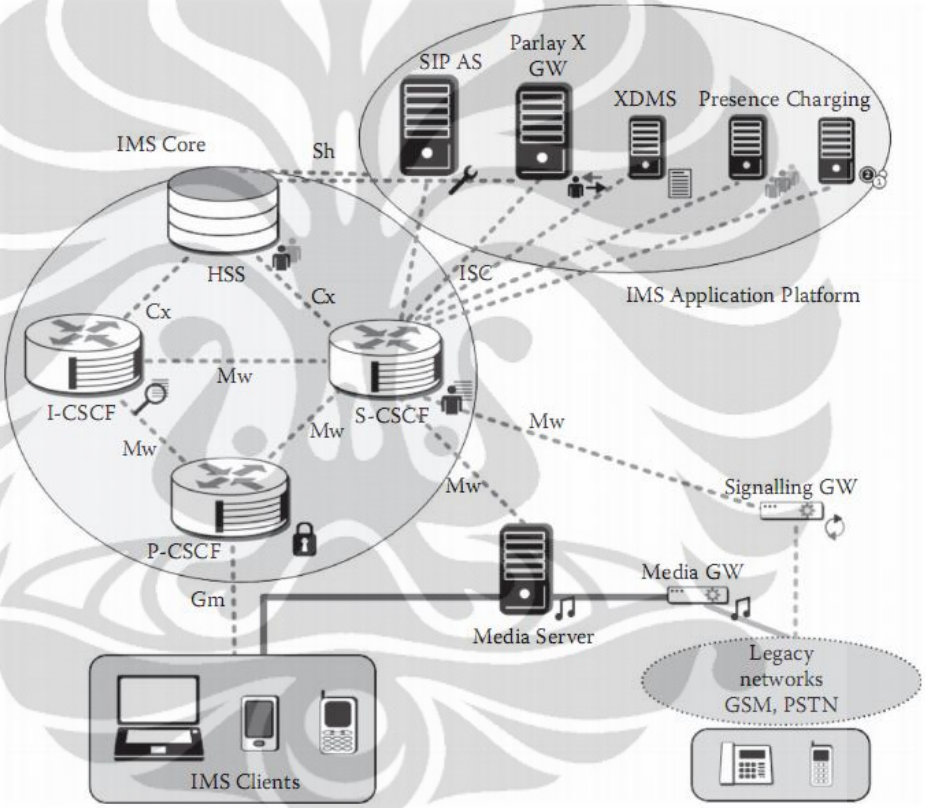
Untuk meluncurkan layanan-layanan baru dan aplikasi-aplikasi dengan cepat, operator dapat menggunakan IMS untuk menghilangkan kompleksitas *platform* layanan yang berbeda dalam jaringan. Dengan adanya layanan konvergensi, maka konsumen dapat menggunakan berbagai macam layanan dari berbagai jaringan dengan data-data pribadi yang sama pada setiap layanannya dan pembukuan penggunaan layanan-layanan tersebut dapat tercantum dalam satu rekening.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

2.2 *IP Multimedia Subsystem* (IMS)

2.2.1 Arsitektur IMS

Sebagai salah satu *platform* yang menyediakan konvergensi telekomunikasi, *Internet Protocol Multimedia Subsystem* (IMS) merupakan suatu *platform* berbasis IP, yang menyatukan berbagai macam *platform-platform* layanan yang ada sehingga menjadi lebih mudah untuk diaskes.



Gambar 2.2 Arsitektur IMS [4]

Berdasarkan Gambar 2.2, terlihat bahwa pada arsitektur IMS terdapat bagian-bagian seperti *IMS Clients*, *IMS Application Platform*, *IMS Core*, dan lain-lain [4]. Komponen-komponen dan bagian-bagian terpenting dari arsitektur IMS tersebut antara lain:

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

a. *Proxy Call State Control Function* (P-CSCF)

Proxy Call State Control Function (P-CSCF) merupakan titik masuk atau gerbang pertama untuk memasuki jaringan IMS, yang berada di dalam *IMS Core*. Semua proses pensinyalan SIP dari dan ke UE *(user equipment)* melintasi P-CSCF. P-CSCF berfungsi seperti sebuah *proxy*, yang menerima dan meneruskan segala permintaan dan respon dari *user*. Hanya *end point* yang terdaftar, yang diizinkan masuk ke dalam jaringan IMS.

b. *Interrogating Call State Control Function* (I-CSCF)

Interrogating Call State Control Function (I-CSCF) merupakan titik hubung pertama untuk memasuki jaringan dari suatu *operator*. Fungsi utama I-CSCF adalah untuk mengalokasikan S-CSCF yang tepat untuk *user*. Pada saat *user* melakukan registrasi, P-CSCF akan meneruskan permintaan dan respon SIP ke I-CSCF. Kemudian, I-CSCF akan menghubungi HSS untuk mendapatkan alamat S-CSCF yang sesuai dengan permintaan *user*.

c. *Serving Call State Control Function* (S-CSCF)

Serving Call State Control Function (S-CSCF) merupakan gerbang terakhir sebelum *user* dapat menggunakan layanan. S-CSCF menyediakan akses kepada *user* untuk menggunakan layanan sesuai dengan permintaan. S-CSCF memegang peranan penting dalam suatu jaringan *operator*. Hal ini dikarenakan S-CSCF yang mengatur hubungan antara *user* dengan *platform* suatu layanan untuk bisa terhubung dengan layanan yang ingin digunakan.

d. *Home Subscriber Server* (HSS)

Home Subscriber Server (HSS) serupa dengan HLR (*Home Location Register*) pada jaringan 2G. HSS merupakan database utama dari IMS yang menyimpan data-data pengguna IMS maupun profil dari aplikasi *server*.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

e. *Application Server* (AS)

Application Server (AS) menyediakan *platform* layanan pada lingkungan IMS.

f. *Media Resource Function* (MRF)

Media Resource Function (MRF) dapat dibagi menjadi dua bagian, yaitu *Media Resource Function Controller* (MRFC) dan *Media Resource Function Processor* (MRFP). MRF menyediakan *media stream processing* seperti *media mixing, media announcements*, *media analysis*, dan *media transcoding.*

g. *IMS end-user system*

IMS end-user system menyediakan dukungan protokol IMS yang diperlukan (SIP) dan media yang terkait dengan pelayanan codec untuk aplikasi multimedia di samping dukungan konektivitas dasar (misalnya, GPRS dan WLAN).

2.2.2 Layanan pada IMS

Ada beberapa macam layanan yang disediakan oleh IMS, antara lain:

1. IM (*Instant Messaging*)

   IM (*Instant Messaging*) merupakan suatu layanan yang memungkinkan pengguna layanan untuk melakukan suatu percakapan (*chatting*) dengan pengguna layanan lainnya. Selain itu, layanan ini juga bisa digunakan untuk melakukan pertukaran data, link, bahkan bisa digunakan untuk menelpon.

2. IPTV (*Internet Protocol Television*)

   IPTV (*Internet Protocol Television*) merupakan suatu layanan *internet television* yang menyiarkan acara TV ataupun video melalui IP (*Internet Protocol*).

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

3. VoD (*Video on Demand*)

   VoD (*Video on Demand*) merupakan suatu layanan yang memungkinkan pengguna layanan untuk memilih dan menonton video sesuai dengan keinginannya. Untuk layanan satu ini, pengguna layanan dapat memilih dan menonton kapan saja serta dapat melakukan *fast forward, rewind, stop, pause, play, dan lainnya*, seperti layaknya menonton VCD/DVD.

4. VoIP (*Voice over Internet Protocol*)

   VoIP (*Voice over Internet Protocol*) merupakan suatu layanan yang memungkinkan pengguna layanan untuk melakukan percakapan jarak jauh melalui media internet atau jaringan berbasis IP (*Internet Protocol*).

2.3 *Video on Demand* (VoD)

Seperti yang telah dijelaskan sebelumnya, *Video on Demand* (VoD) merupakan suatu layanan yang memungkinkan pengguna layanan untuk memilih dan menonton video sesuai dengan keinginannya. Selain itu, pengguna layanan juga bisa kapan saja menonton video pilihan mereka sendiri. Ada tiga macam VoD [5], antara lain:

1. *Near Video On Demand* (NVOD)

   NVOD dijalankan oleh televisi berbasis *cable* dan *satellite*. Sistem ini memungkinkan seseorang untuk melakukan *pay-per-view* program yang dikeluarkan oleh *multiple-broadcasters*. Ini membuat konsumen tidak lagi terikat waktu untuk menyaksikan acara yang ia inginkan.

2. *Push Video On Demand* (PVOD)

   Ini sebenarnya lebih merupakan duplikasi kurang sempurna dari konsep utama VOD itu sendiri. Karena pada dasarnya PVOD menawarkan hal yang sama dengan VOD tetapi justru dengan lebih

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

banyak kekurangan. Misalnya dari sisi *memory*. Program yang bisa konsumen unduh hanya bertahan seminggu karena keterbatasan *memory* ini.

3. *Manufacturing on Demand* (MOD)

   Manufacturing on Demand (MOD) dikenal juga dengan nama *DVD on Demand*. MOD ini mendekati konsep DVD karena konsumen bisa memiliki perangkat keras dari apa yang ia inginkan. Bentuknya bisa berupa DVD. Ini menjadi pilihan bagi perusahaan pembuat film ataupun serial televisi yang memiliki sesuatu yang diprediksi tidak akan begitu laku di pasaran. Namun jika ada copy digitalnya, mereka tetap bisa menjual kepada orang-orang yang menginginkannya saja.

2.4 Protokol-protokol untuk Layanan Multimedia

Berikut ini ada beberapa protokol untuk layanan multimedia [6], yaitu:

a. RSVP (*Resource Reservation Protocol*)

   RSVP (*Resource Reservation Protocol*) merupakan sebuah protokol yang digunakan untuk memesan atau menyediakan bandwith sehingga data dapat terkirim sampai tujuan dengan cepat dan tepat.

b. SMRP (*Simple Multicast Routing Protocol*)

   SMRP (*Simple Multicast Routing Protocol*) merupakan sebuah protokol yang mendukung *conferencing* dengan menggandakan data pada sekelompok penerima. Jadi, sekelompok penerima tersebut menerima data secara bersamaan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

c. RTSP (*Real-Time Streaming Protocol*)

RTSP (*Real-Time Streaming Protocol*) merupakan sebuah protokol di level aplikasi yang berfungsi untuk membangun dan mengontrol pengiriman data secara *real-time*. Pada umumnya digunakan pada *Video on Demand* (VoD). RTSP memiliki beberapa perintah yang dikirimkan *client* ke *server streaming* RTSP, yaitu:

- *Setup*

  Server akan mengalokasikan sumber daya kepada sesi *client.*

- *Play*

  Server akan mengirimkan sebuah *stream* ke sesi *client* yang telah dibangun berdasarkan perintah *setup.*

- *Pause*

  Server akan menunda pengiriman sebuah *stream* dan tetap menjaga sumber daya yang telah dialokasikan.

- *Teardown*

  Server akan memutuskan hubungan yang telah terbangun dan membebas-tugaskan sumber daya yang telah dialokasikan.

d. RTP (*Real-Time Transport Protocol*)

RTP (*Real-Time Transport Protocol*) merupakan sebuah protokol yang berfungsi untuk mengirimkan data multimedia secara *real-time* dan bergantung pada protokol Transport.

e. RTCP (*Real-Time Control Protocol*)

RTCP (*Real-Time Control Protocol*) merupakan sebuah protokol yang berfungsi untuk menjamin kualitas *streaming* atau

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

dapat dikatakan sebagai *Protocol QoS* (*Quality of Service*). RTCP merupakan bagian pengkontrolan paket data pada RTP.

f. SIP (*Session Initiation Protocol*)

SIP (*Session Initiation Protocol*) merupakan *peer to peer signaling* protokol, yang memperbolehkan *endpoint* untuk memulai dan mengakhiri sesi komunikasi [7]. Arsitektur dari SIP terdiri dari dua komponen, yaitu *user agent* dan *server*.

Ada empat jenis *server* pada SIP, antara lain:

- *Proxy Server*

  Proxy Server merupakan *host* jaringan yang berperan sebagai perantara yang bertujuan untuk meminta *request* atas nama *client* yang lain.

- *Redirect Server*

  Redirect Server merupakan kesatuan logika yang mengarahkan suatu *client* pada perangkat pengganti dari *Uniform Resorce Indocators (URIs)* untuk menyelesaikan tugas *request.*

- *Registrar Server*

  Registrar Server merupakan suatu *server* yang menerima dan memproses pesan pendaftaran yang memperbolehkan laksi dari suatu *endpoint* dapat diketahui keberadaannya. Kerja dari *server* jenis ini berhubungan dengan *Location Server*.

- *Location Server*

  Location Server merupakan *server* yang menyediakan *service* untuk *database* abstrak yang berfungsi mentranslasikan alamat dengan kata atau keterangan yang ada pada domain jaringan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Pada SIP, terdapat dua format *message*, yaitu:

- *Request*

  Request dikirimkan dari *client* ke *server*, berisi tentang operasi yang diminta oleh *client.*

- *Responses*

  Responses dikirim dari *server* ke *client*, berisi informasi mengenai status dari apa yang diminta oleh *client*.

Ada tujuh tipe dari *request messages*, antara lain:

- REGISTER: digunakan oleh *client* untuk mendaftarkan informasi kontak.
- INVITE: menunjukkan bahwa *user* atau *service* sedanng diundang untuk bergabung dalam *session.* Isi dari pesan ini akan memasukkan suatu uraian menyangkut *session* untuk *user* yang akan diundang.
- OPTION: digunakan untuk *query* suatu *server* tentang kemampuan yang dimilikinya.
- CANCEL: digunakan untuk membatalkan suatu *request* yang sedanng menunggu keputusan.
- ACK: mengkonfirmasi bahwa *client* telah menerima suatu final *response* untuk suatu INVITE *request.*
- BYE: dikirim oleh *user agent client* untuk menunjukkan pada *server* bahwa percakapan ingin segera diakhiri.
- PRACK: meningkatkan kehandalan jaringan dengan menambahkan *acknowledgement system* ke *provisional responses* [8].

Response message 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:

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

- 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.

2.5 Sistem *Charging* pada IMS

Charging merupakan sebuah fungsi dalam jaringan operator yang memfasilitasi pengumpulan pendapatan atas penggunaan sumber daya (layanan) [1]. Oleh karena itu, segala proses yang melibatkan *charging, billing,* dan *accounting* merupakan operasi yang paling penting dan sangat diperlukan untuk mendukung jaringan telekomunikasi.

Seperti yang telah distandarisasi oleh 3GPP, *charging* pada IMS terdiri dari beberapa elemen, yaitu:

a. *Charging Trigger Function* (CTF)

Charging Trigger Function (CTF) akan dikerahkan sebagai bagian integral dari setiap *Application Servers* (AS), dimana informasi tentang *charging* akan dikumpulkan.

b. *Charging Data Function* (CDF)

Charging Data Function (CDF) digunakan untuk proses *offline charging,* dan melibatkan pertukaran *Diameter accounting messages* dengan CTF di titik referensi Rf (*Rf reference point*).

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

c. *Online Charging Function* (OCF)

Online Charging Function (OCF) diperlukan untuk proses *online charging.* Pertukaran *Diameter credit control messages* dengan CTF terjadi di *Ro interface.*

Proses *charging* pada IMS dapat dilakukan baik secara *online* maupun *offline.* Pada *online charging,* terjadi interaksi secara *real-time* antara mekanisme pengaturan *charging* dengan konsumsi sumber daya. Sedangkan pada *offline charging,* tidak terjadi interaksi secara *real-time* antara pengaturan *charging* dengan konsumsi sumber daya.

a. *Offline Charging*

Elemen-elemen yang ada pada proses *offline charging,* yaitu: CTF, CDF, *Charging Gateway Function* (CGF), dan *billing domain.* Pada sistem ini, CDF dan CGF sering diintegrasikan ke dalam satu kesatuan dalam *billing domain,* yang dihubungkan dengan Rf *reference.*

CTF terintegrasi dengan elemen-elemen pada IMS, seperti P-CSCF, I-CSCF, S-CSCF, SIP AS, dan lain-lain. CTF menghasilkan dan mengirimkan *Diameter Accounting Requests* (ACR) ke CDF ketika mendeteksi suatu kejadian yang berhubungan dengan *charging.* ACR dikirimkan dalam *Diameter Attribute Value Pairs* (AVP) dari *Diameter messages.*

Ketika awal sesi *charging* terdeteksi, CTF mulai mengumpulkan kejadian-kejadian yang berhubungan dengan *charging* dan mengirimkan informasi yang dikumpulkan dalam *interim* ACR *messages.*

Pada proses penghentian sesi *charging,* ACR terakhir akan dikirimkan ke CDF. *Accounting Answer* (ACA) *messages* akan dikirim oleh CDF ke CTF dalam menanggapi ACR *messages* yang diterima.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

b. *Online Charging*

Protokol yang digunakan pada *online charging* adalah *Diameter Credit Control protocol. Credit control* diperlukan untuk sesi pra-otorisasi sebelum dimulainya penggunaan sumber daya. Seperti pada *offline charging, online charging* memerlukan CTF dan OCF sebagai tambahannya, dimana OCF merupakan bagian dari *Online Charging System* (OCS).

Pada *online charging,* CTF terintegrasi dengan elemen-elemen pada IMS, seperti MRCF, SIP AS, dan S-CSCF. Diperlukan *IMS gateway* untuk mendukung *credit control* pada S-CSCF. Komunikasi antara CTF dan OCF terjadi pada *Ro interface,* yang mendukung *Diameter credit control.* Ketika kejadian-kejadian yang berhubungan dengan *charging* terdeteksi, *Credit Control Request* (CCR) *message* dikirim ke OCS. Fungsi pentarifan dalam OCS akan menentukan nilai dari sumber daya yang diminta untuk mengaktifkan alokasi dari suatu jumlah tertentu dari sumber daya yang digunakan, misalnya lama pemakaian dan lain-lain. *Credit* yang dialokasikan akan dikirim ke CTF dalam sebuah *Diameter Credit Control Answer* (CCA).

Dikarenakan oleh penggunaan sumber daya, fungsi manajemen keseimbangan *account* (*account balance management function*) akan melakukan pengawasan terhadap penggunaan *credit* untuk memberikan sinyal penghentian layanan jika *credit* habis.

Selain itu, pada IMS juga terdapat beberapa jenis *charging*, antara lain:

1) *Session Based Charging*

Pada jenis ini, dilakukan *charging* berdasarkan sesi yang terjadi.

- *Duration Charging*

  Proses *charging* dilakukan berdasarkan lamanya penggunaan layanan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

2) *Event Based Charging*

Event Based Charging digunakan untuk setiap transaksi tunggal antara *end user* dengan jaringan, misalnya permintaan layanan VoD.

- *Content Charging*

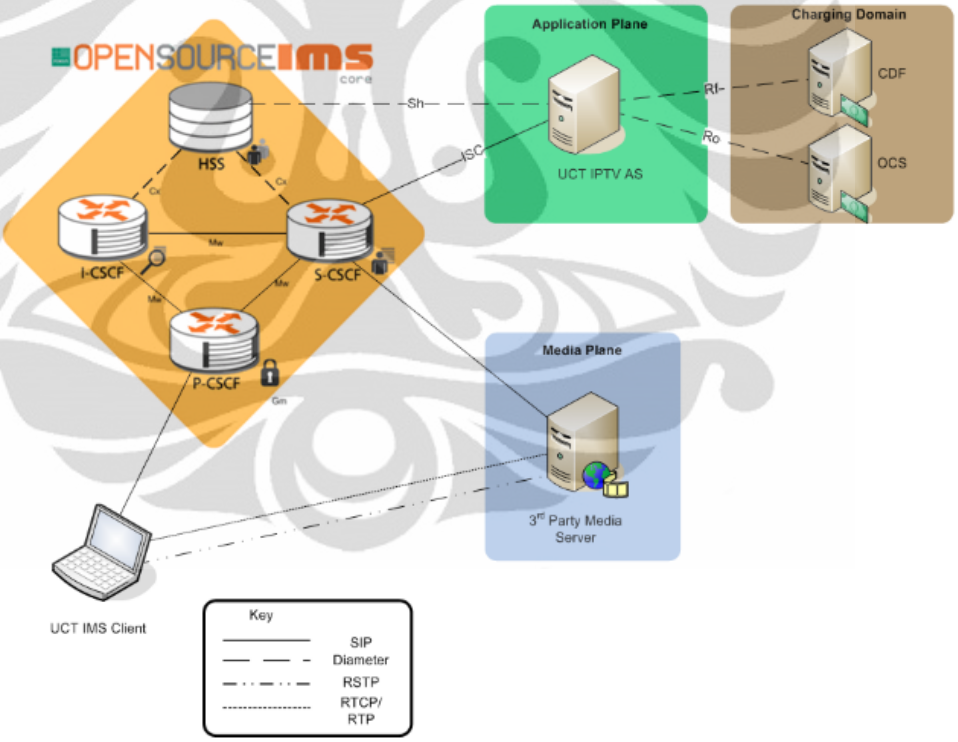
  Jenis *charging* ini sangat diperlukan bagi penyedia layanan dan aplikasi. Proses *charging* dilakukan berdasarkan setiap *content* yang digunakan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 3
PERANCANGAN SISTEM *CHARGING* UNTUK LAYANAN IPTV (VOD)

3.1 Skenario Uji Coba

Pada skripsi ini, dilakukan uji coba terhadap layanan *Video on Demand* (VoD) dengan menggunakan *Open Source IMS*. Ada dua hal yang menjadi tujuan utama dari penelitian ini, yaitu menganalisis *message flow* ketika *client* ingin menggunakan layanan *Video on Demand* dan mengetahui dampak yang ditimbulkan terhadap *client* dengan adanya implementasi *UCT IMS Charging System*. Berikut ini adalah arsitektur dari *IPTV* (VoD) *charging*:



Gambar 3.1 Arsitektur *IPTV (VoD) Charging* [9]

Berdasarkan Gambar 3.1, maka arsitektur *IPTV (VoD) charging* meliputi tahap-tahap berikut:

Universitas Indonesia
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Tahap Pertama : *UCT IMS Client* mengirimkan permintaan INVITE pada salah satu saluran kemudian mengirimnya pada *Indirection AS.*

Tahap Kedua : *Indirection AS* kemudian mencocokkan permintaan dengan alamat RTSP yang tersedia, lalu memberikan alamat tersebut pada *UCT IMS Client*.

Tahap Ketiga : Sistem *charging* memicu proses *charging* dan mengirimkan sebuah permintaan ke *offline* atau *online charging functions*.

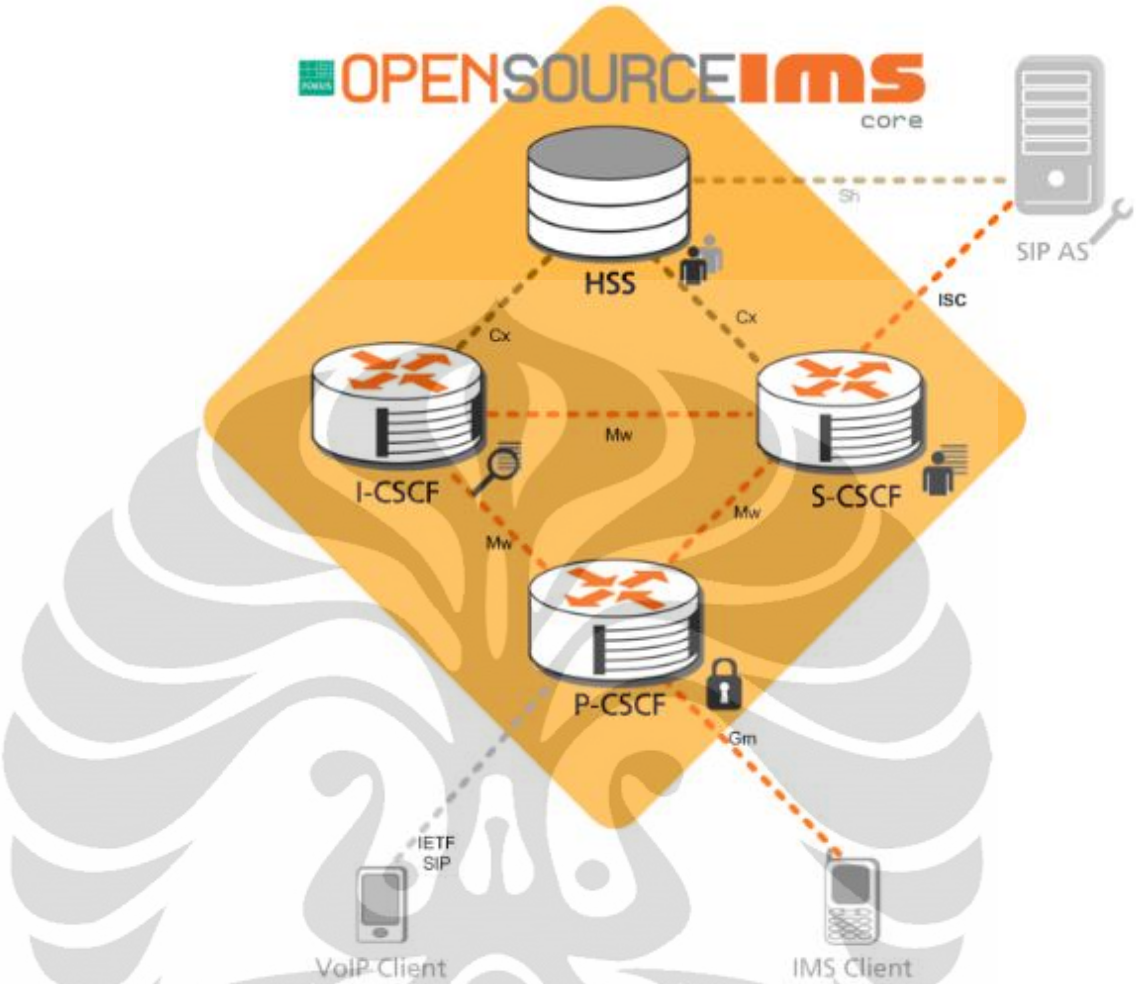
Tahap Keempat : *Charging* dimulai dan IPTV server mengembalikan alamat RTSP yang relevan ke *UCT IMS Client* dalam respon 200 OK.

Tahap kelima : *UCT IMS Client* mulai menginisiasi sesi RTSP dengan *media server* pihak ketiga.

3.2 *OpenIMSCore*

Perancangan sistem *charging* untuk layanan IPTV (VoD) ini menggunakan *OpenIMSCore* yang dikembangkan oleh Fraunhofer FOKUS. *OpenIMSCore* merupakan suatu implementasi *Open Source* dari *Call Session Control Function* (CSCFs) dan *Home Subscriber Server* (HSS) pada IMS, yang bersama-sama membentuk elemen-elemen inti dari seluruh arsitektur IMS/NGN [10].

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 3.2 *OpenIMSCore* [10]

3.3 *Application Server*

Selain *OpenIMSCore*, pada perancangan ini juga dibutuhkan *application server* [11]. Fungsi dari *application server* adalah mengarahkan r*equest* dari *client* ke alamat dari *media server* yang dituju. Oleh karena itu, *application server* yang akan digunakan pada perancangan ini disebut *indirection application server*. Adapun langkah-langkah untuk membangun sebuah *application server*, adalah sebagai berikut:

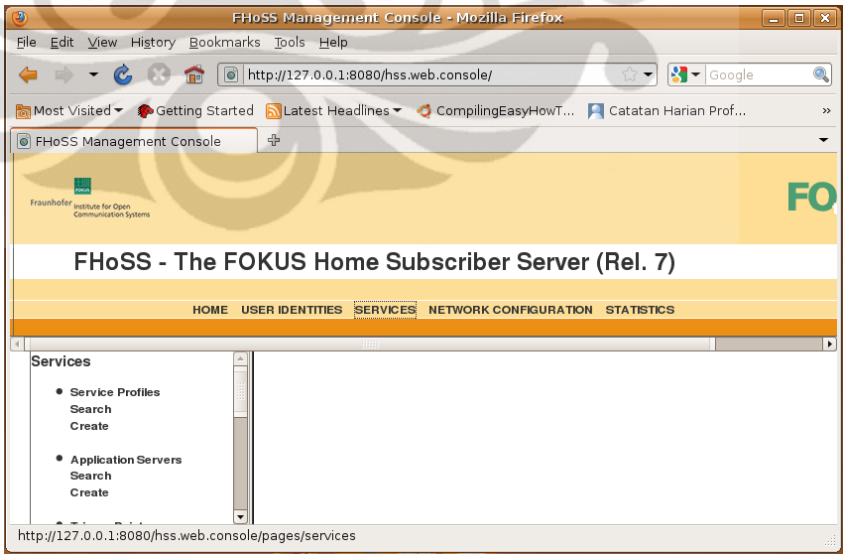
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

1. *Download* dan *Install Application Server*

   Download *server*-nya dari https://developer.berlios.de/project/showfiles.php?group_id=7844, kemudian *install*. Ada dua jenis package yang disediakan, yaitu .deb dan .tar.gz. Apabila menggunakan .deb, maka untuk menginstalnya hanya perlu mengklik *file* tersebut dua kali lalu klik *install*. Sedangkan apabila menggunakan .tar.gz, maka untuk menginstalnya dilakukan dengan meng-*extract* terlebih dahulu *file* tersebut dan kemudian mengetik "*make*" pada terminal dari direktori root *file* yang telah di-*extract* tadi.

2. Melakukan Konfigurasi FHoSS

   Setelah menginstal *application server* di atas, buka alamat http://127.0.0.1:8080 atau http://IP_address_server_IMS:8080 dengan menggunakan *web browser* apapun. Kemudian *login* dengan *username* dan *password* yang telah disampaikan pada bagian sebelumnya. Setelah *login*, lalu pilih menu "*SERVICES*" dan ikuti langkah-langkah berikut:

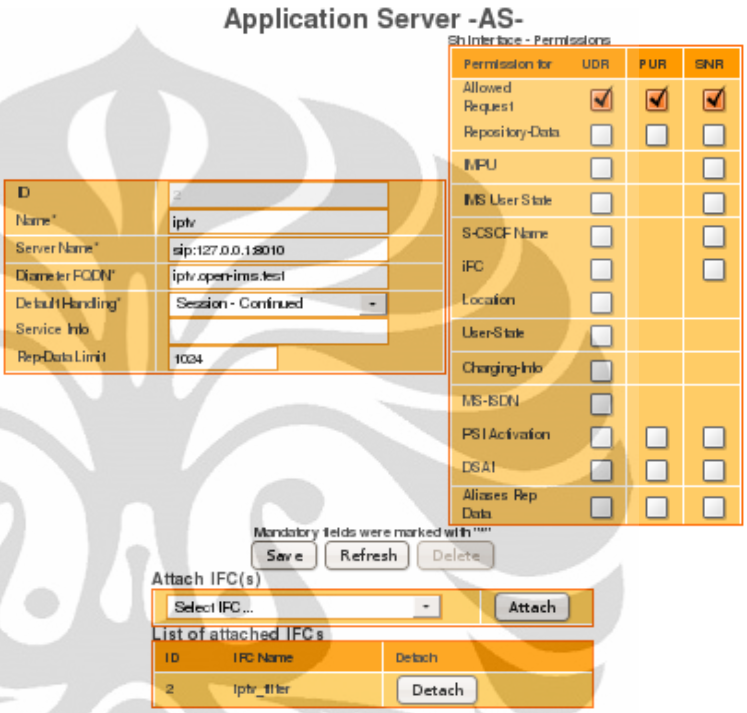


Gambar 3.3 Tampilan *FHoSS – SERVICES*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

a. Membuat *Application Server*

Setelah masuk pada bagian "*SERVICES*", pilih bagian "*Application Servers*" lalu pilih "*Create*" dan isi *form* sesuai dengan yang tertera pada Gambar 3.4.



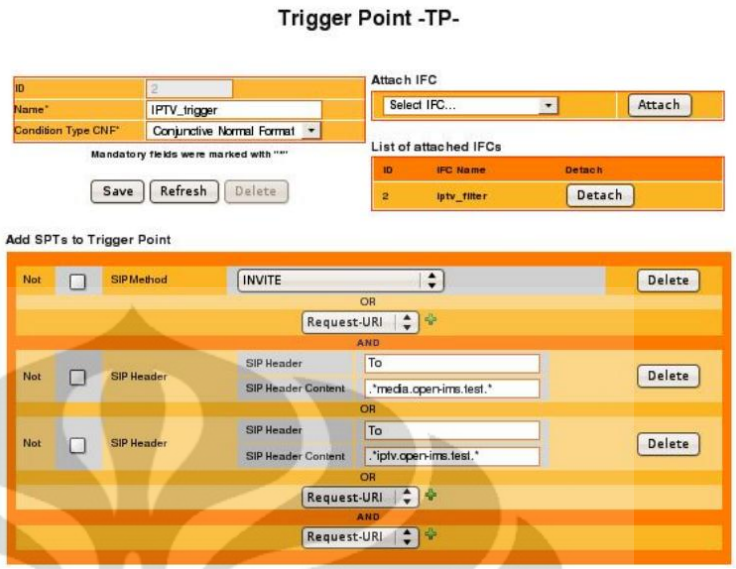
Gambar 3.4 Konfigurasi *Application Server*

Pada bagian ini, telah dibuat sebuah *application server* yang dikenal dengan nama *UCT Indirection Application Server* pada server IMS. *Application server* ini diberi nama "iptv" dan terletak pada alamat IP 127.0.0.1 (*localhost*) dengan port 8010.

b. Membuat *Trigger Point*

Langkah selanjutnya adalah membuat *Trigger Point* untuk *application server* yang telah dibuat tadi. Pilih bagian "*Trigger Points*" lalu pilih "*Create*" dan isi *form* sesuai dengan yang tertera pada Gambar 3.5.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 3.5 Konfigurasi *Trigger Point*

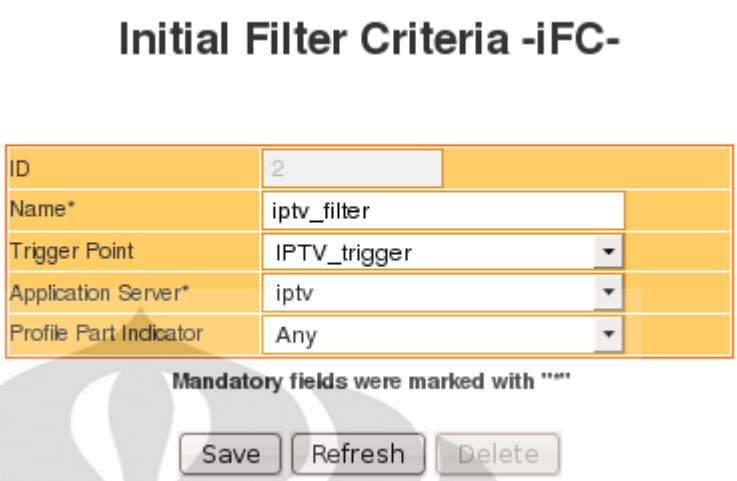
Pada bagian ini, dibuat sebuah *trigger point* yang berfungsi untuk memulai inisiasi suatu bentuk permintaan yang datang pada server IMS, contohnya pada IPTV akan ditentukan permintaan mana saja yang digolongkan sebagai permintaan akan layanan IPTV.

Trigger point yang diberi nama "*IPTV_trigger*" ini akan menggolongkan segala bentuk permintaan yang ada berdasarkan pada bagian "*Add SPTs to Trigger Point*".

c. Menghubungkan *Application Server* dan *Trigger Point* dengan *Initial Filter Criteria* (iFC)

Setelah membuat *application server* dan *trigger point*, langkah selanjutnya adalah menghubungkan keduanya. Pilih bagian "*Initial Filter Criteria*" lalu pilih "*Create*" dan isi *form* sesuai dengan yang tertera pada Gambar 3.6.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 3.6 Konfigurasi *Initial Filter Criteria*

Dengan adanya *Initial Filter Criteria (iFC)* ini, maka setiap permintaan akan layanan IPTV yang ada akan langsung diarahkan ke server IPTV (*UCT Indirection Application Server*) untuk diproses lebih lanjut.

d. Tambahkan *Initial Filter Criteria (iFC)* pada *Shared iFC Sets*

Setelah membuat *Initial Filter Criteria* (iFC), langkah selanjutnya adalah menambahkan *Initial Filter Criteria* (iFC) pada *Shared iFC Sets*. Klik "*Search*" pada bagian "*Shared iFC Sets*" lalu pilih "*default_shared_set*" dan isi *form* sesuai dengan yang tertera pada Gambar 3.7.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

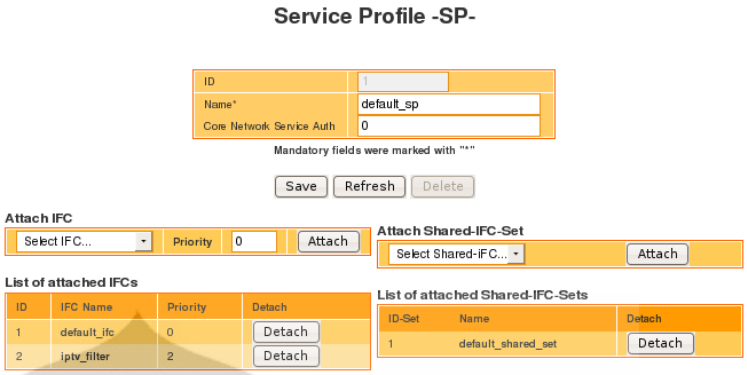
Gambar 3.7 Konfigurasi *Shared iFC Sets*

Pada bagian ini, filter-filter yang telah dibuat sebelumnya seperti *iptv_filter* dimasukkan ke dalam "*List of attached IFCs*". Dengan demikian, seluruh pelanggan dapat mengaksesnya.

e. Memasukkan *Initial Filter Criteria (iFC)* dan *Shared iFC Sets* ke *Service Profile*

Ini merupakan langkah terakhir pada FHoSS. Klik "*Search*" pada bagian "*Service Profiles*" lalu pilih "*default_sp*" dan isi *form* sesuai dengan yang tertera pada Gambar 3.8.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 3.8 Konfigurasi *Service Profile*

3. Mengatur *file key_value_file.xml*

Pada tahap ini, *file* xml pada *UCT Indirection Application Server* harus sesuai dengan alamat dari *media server* yang dituju. Cara mengaturnya adalah sebagai berikut:

#sudo gedit /usr/share/uctiptv_advanced/key_value_file

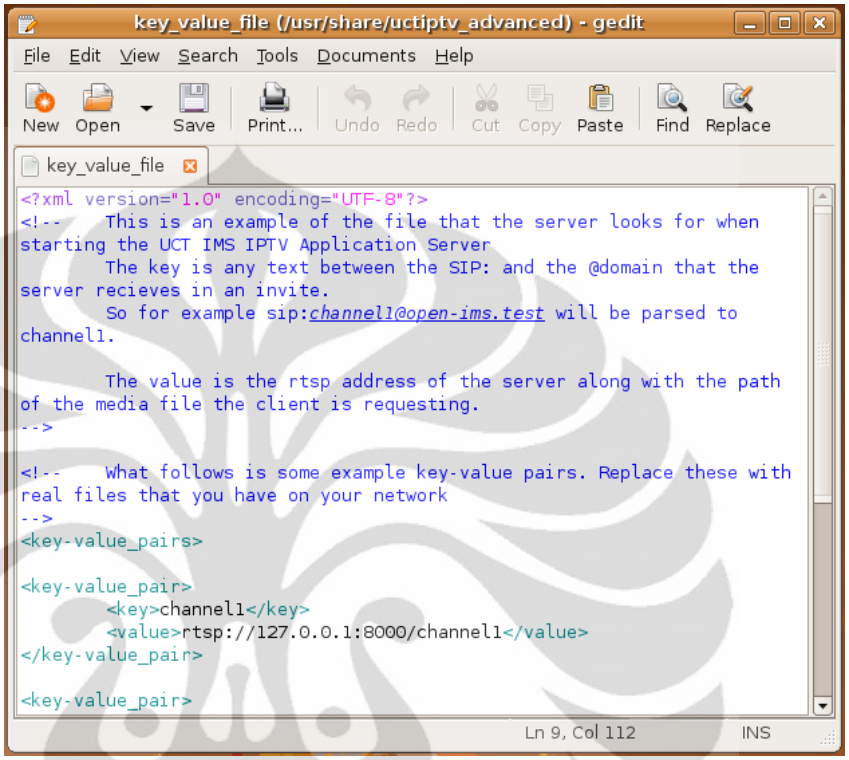
Setelah mengetikkan *command* di atas pada terminal, maka akan muncul *default key_value_file* seperti di bawah ini:

```
<?xml version="1.0" encoding="UTF-8"?>
<key-value_pairs>
   <key-value_pair>
      <key>channel1</key>
         <value>rtsp://media_server_address.domain:8000
                                 /requested_channel</value>
   </key-value_pair>
</key-value_pairs>
```

Kemudian, ubah alamat RTSP di atas dengan alamat RTSP yang digunakan. Pada skripsi ini digunakan alamat RTSP 127.0.0.1

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

dengan port 8000, sehingga *key_value_file* berubah menjadi seperti pada Gambar 3.9.



Gambar 3.9 Tampilan *key_value_file*

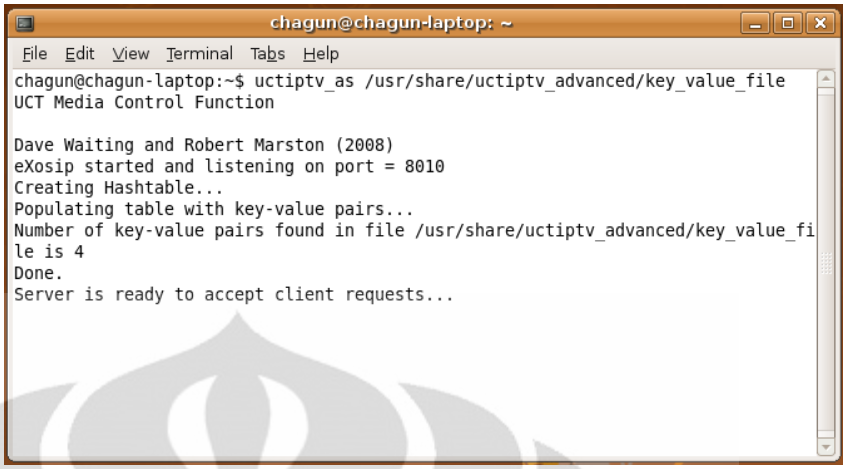
4. Menjalankan *UCT Indirection Application Server*

Setelah mengatur *file key_value_file.xml,* jalankan *UCT Indirection Application Server* dengan mengetikkan *command* berikut pada terminal:

#uctiptv_as /usr/share/uctiptv_advanced/key_value_file

Apabila berhasil, maka akan seperti pada Gambar 3.10.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 3.10 Tampilan *UCT Indirection Application Server*

3.4 *Media Server*

Media server berfungsi untuk men-*streaming*-kan video atau film yang akan diminta oleh *client*. Pada perancangan ini, *media server* yang digunakan adalah dengan VLC. Pertama-tama, *download* dan *install* VLC beserta *plugin-plugin*-nya. Caranya adalah dengan mengetik *command* berikut pada terminal:

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

Setelah itu, buat *file* konfigurasi dari video atau film yang akan di-*streaming*-kan dengan nama *vod.vlm*. Caranya adalah dengan mengetikkan *command* di bawah ini pada terminal:

#sudo gedit vod.vlm

Kemudian masukkan kalimat seperti di bawah ini pada *file* tersebut:

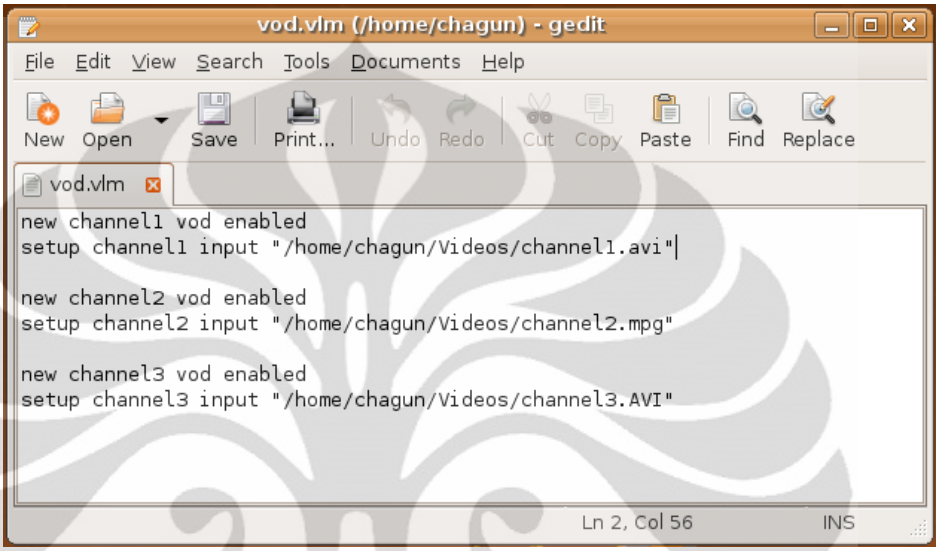
new requested_channel vod enabled
setup requested_channel input "letak video/film"

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

keterangan:

- ubah *requested_channel* dengan nama yang diinginkan.
- masukkan alamat video/film yang ingin di-*streaming*-kan.

Pada skripsi ini, isi dari *vod.vlm* adalah sebagai berikut:



Gambar 3.11 Tampilan *vod.vlm*

Apabila langkah di atas telah dilakukan, maka video/film pun siap untuk di-*streaming*. Caranya adalah dengan mengetikkan *command* berikut pada terminal:

#vlc –vvv --ttl 12 –intf telnet --rtsp-host 127.0.0.1:8000 --vlm-conf vod.vlm

keterangan:

- 127.0.0.1:8000 bisa diubah sesuai dengan alamat RTSP yang digunakan pada *key_value_file*.

3.5 Pengetesan IPTV (VoD)

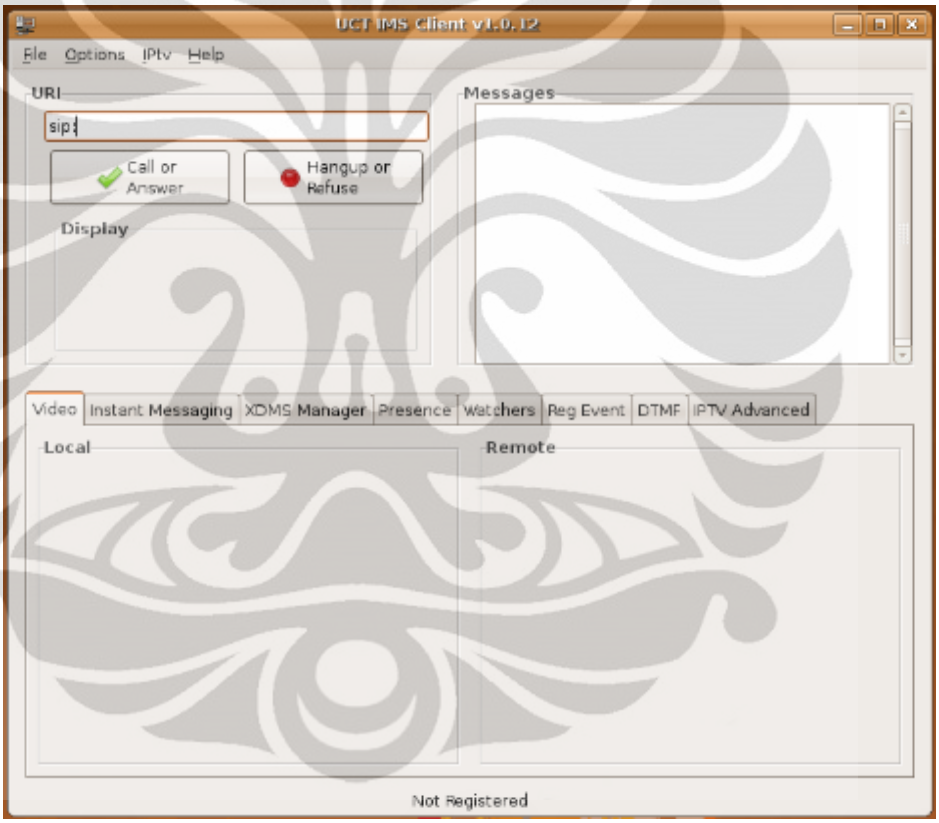
Langkah pertama yang harus dilakukan adalah menginstal *software* yang akan digunakan sebagai *client*. Pada skripsi ini, digunakan *UCT IMS Client* yang dapat diperoleh dari

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

https://developer.berlios.de/project/showfiles.php?group_id=7844. Setelah berhasil men-*download* dan menginstalnya, *UCT IMS Client* siap untuk digunakan. Cara menjalankannya adalah dengan mengetik *command* berikut pada terminal:

#sudo uctimsclient

Apabila berhasil, maka akan muncul seperti Gambar 3.12.



Gambar 3.12 Tampilan *UCT IMS Client*

Setelah itu, *register* dengan sebagai Alice atau Bob dengan cara klik "*Options*" lalu klik "*Register as Alice*" atau "*Register as Bob*". Setelah berhasil melakukan *register*, maka *client* sudah dapat menikmati layanan IPTV (VoD) dengan cara mengklik "IPtv" dan memilih "*Channel 1*", "*Channel 2*", atau "*Channel 3*". Apabila berhasil, maka video/film pun

akan muncul. Dengan catatan, *application server* dan *media server* harus sudah dijalankan terlebih dahulu.

Pada skripsi ini akan dicontohkan dengan *client* memilih "*Channel 2*" dengan alamat sip:channel2@open-ims.test, dan hasilnya sebagai berikut:

Gambar 3.13 Tampilan IPTV (VoD)

3.6 *UCT IMS Charging System*

Bagian terakhir dari perancangan pada skripsi ini adalah *UCT IMS Charging System*, dimana bagian ini berfungsi untuk melakukan proses *charging* ketika *client* menggunakan layanan VoD.

Adapun langkah-langkah untuk membangunnya adalah sebagai berikut:

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

a. *Download* dan *install UCT IMS Charging System*

UCT IMS Charging System dapat di-*download* di https://developer.berlios.de/project/showfiles.php?group_id=7844. Setelah itu, *install file* tersebut.

Ada dua tipe *file*, yaitu .deb.tar.gz dan .tar.gz. Apabila menggunakan tipe *file* .deb.tar.gz, maka untuk menginstalnya hanya dengan meng-*extract file* tersebut dan mengeklik masing-masing *file* sebanyak dua kali lalu klik *install*. Sedangkan apabila menggunakan tipe *file* .tar.gz, maka untuk menginstalnya harus di-*extract* terlebih dahulu. Setelah itu, buka masing-masing *file* melalui terminal dan ketikkan "*make*" pada terminal dari masing-masing direktori rootnya.

Pada masing-masing penginstalan, diperlukan *package-package* tertentu. Untuk tipe *file* .deb.tar.gz, diperlukan *package* libosip dan libexosip. Sedangkan untuk tipe *file* .tar.gz, diperlukan *package* libosip (3.0.3), libeXosip (3.0.3), libosip-dev, dan libexosip-dev.

b. Atur FHoSS, seperti pada bagian 3.3 (*Application Server*).

c. Setelah *point* a dan b telah dilakukan, langkah selanjutnya adalah menjalankan *IPTV Charging System*, *CDF*, dan *OCS*. Caranya adalah sebagai berikut:

- Apabila menggunakan tipe *file* .deb.tar.gz
  Masuk ke dalam root *file IPTV Charging System*, *CDF*, dan *OCS* terinstal. Untuk tipe ini terletak pada /usr/share/uctimscharging/. Kemudian ketikkan *command* berikut pada masing-masing tab terminal:

  ./uctimscharging-iptv (untuk *IPTV Charging Serever*)

  ./uctimscdf (untuk *CDF*)

  ./uctimsocs (untuk *OCS*)

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

- Apabila menggunakan tipe *file* .tar.gz
  Masuk ke dalam root *file IPTV Charging System*, *CDF*, dan *OCS* terinstal. Untuk tipe ini, *file* terletak sesuai pada saat proses penginstalan. Kemudian ketikkan *command* berikut pada masing-masing tab terminal:

  ./uctimscharging-iptv (untuk *IPTV Charging Serever*)

  ./uctimscdf (untuk *CDF*)

  ./uctimsocs (untuk *OCS*)

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 4
HASIL UJI COBA DAN ANALISIS

Setelah merancang dan melakukan uji coba, dilakukan analisis *message flow* mulai dari ketika *client* melakukan registrasi sampai dengan *client* tersebut melakukan deregistrasi. Selain itu juga dilakukan analisis implementasi *UCT IMS Charging System* untuk layanan *Video on Demand.*

4.1 Analisis *Message Flow*

Ada empat jenis *message flow* yang akan dianalisis pada uji coba ini, yaitu *message flow* pada saat registrasi, pada saat meminta layanan *Video on Demand,* pada saat pemutusan layanan *Video on Demand,* dan pada saat deregistrasi.

4.1.1 *Message Flow* pada Saat Registrasi

Uji coba ini dilakukan dengan registrasi *client* dengan IMPU sip:alice@open-ims.test, sehingga diperoleh *message flow* seperti berikut:

Tabel 4.1 *Message flow* pada saat 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 (0 bindings)
3.	*Client*	*IMS Core*	SIP	Request: REGISTER sip:open-ims.test
4.	*IMS Core*	*Client*	SIP	Status: 200 OK - SAR successful and registrar saved (1 bindings)

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

5.	*Client*	*IMS Core*	SIP	Request: SUBSCRIBE sip:alice@open-ims.test
6.	*IMS Core*	*Client*	SIP	Status: 200 Subscription to REG saved
7.	*IMS Core*	*Client*	SIP	Request: NOTIFY sip:alice@10.0.2.15:5061
8.	*Client*	*IMS Core*	SIP	Status: 200 OK

Berdasarkan Tabel 4.1, dapat dianalisis bahwa *message flow* untuk proses registrasi *client* adalah sebagai berikut:

1. Tahap pertama, *client* melakukan *request* untuk memasuki *IMS Core* dengan cara mengirimkan SIP *request REGISTER* sip:open-ims.test.
2. Setelah itu, *IMS Core* membalas pesan tersebut dengan status: *401 Unauthorized – Challenging the UE (0 bindings)*. Hal ini menandakan bahwa pesan *request* yang dikirimkan oleh *client* tidak lengkap. Ketidaklengkapan ini terjadi pada *message header*, tepatnya pada bagian *Authorization.*
3. Setelah melengkapi pesan tersebut, *client* mengirimkan pesan *request REGISTER* kembali ke *IMS Core*.
4. Pada tahap ini, *IMS Core* telah menerima permintaan registrasi si *client*. Hal ini ditandakan dengan balasan pesan yang telah dikirimkan *IMS Core* ke *client* dengan status: *200 OK – SAR successful and registrar saved.*
5. Setelah proses registrasi berhasil, kemudian *client* mengirimkan pesan *request SUBSCRIBE* ke *IMS Core*. Adapun tujuan dari pengiriman pesan tersebut adalah untuk menggunakan layanan-layanan dari IMS yang ada.
6. Apabila berhasil, maka *client* akan menerima pesan balasan dengan status: *200 Subscription to REG saved*.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

7. Setelah semua *request* dari *client* terpenuhi, maka *IMS Core* akan mengirimkan pesan *request NOTIFY*. Adapun tujuan dari pengiriman pesan tersebut adalah untuk memastikan bahwa *request client* telah terpenuhi.
8. Apabila *request client* telah terpenuhi, maka *client* akan mengirimkan pesan balasan dengan status: 200 OK.

4.1.2 *Message Flow* pada Saat Meminta Layanan *Video on Demand*

Uji coba ini dilakukan dengan meminta layanan *Video on Demand (VoD)* dengan alamat sip:channel2@open-ims.test, sehingga diperoleh *message flow* seperti berikut:

Tabel 4.2 *Message flow* pada saat meminta layanan VoD

No.	Source	Destination	Protocol	Info
1.	*Client*	*IMS Core*	SIP	Request: INVITE sip:channel2@iptv.open-ims.test, with session description
2.	*IMS Core*	*Client*	SIP	Status: 100 trying -- your call is important to us
3.	*IMS Core*	*Application Server*	SIP	Request: INVITE sip:channel2@iptv.open-ims.test, with session description
4.	*Application Server*	*IMS Core*	SIP	Status: 100 Trying
5.	*Application Server*	*IMS Core*	SIP	Status: 101 Dialog Establishement
6.	*IMS Core*	*Client*	SIP	Status: 101 Dialog Establishement

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

7.	*Application Server*	*IMS Core*	SIP	Status: 200 OK
8.	*IMS Core*	*Client*	SIP	Status: 200 OK
9.	*Client*	*IMS Core*	SIP	Request: ACK sip:channel2@10.0.2.15:8010
10.	*IMS Core*	*Application Server*	SIP	Request: ACK sip:channel2@10.0.2.15:8010

Berdasarkan Tabel 4.2, dapat dianalisis bahwa *message flow* untuk proses meminta layanan VoD adalah sebagai berikut:

1. *Client* akan mengirimkan *request INVITE* ke *IMS Core* untuk meminta salah satu layanan VoD.
2. *IMS Core* mengirimkan pesan balasan dengan status: *100 trying your call is important to us.* Ini menandakan bahwa *request INVITE* telah diterima oleh *IMS Core* dan sedang diproses.
3. *IMS Core* meneruskan pesan *request* tersebut ke *Application Server* untuk mengecek apakah layanan VoD yang diminta itu ada atau tidak.
4. *Application Server* melakukan pengecekan untuk layanan VoD yang diminta dan membalas pesan *request* yang dikirimkan tadi dengan status: 100 Trying.
5. *Application Server* telah memastikan bahwa layanan VoD yang diminta tersedia dan siap untuk digunakan. Kemudian *Application Server* mengirimkan pesan ke *IMS Core* dengan status: *101 Dialog Establishement*.
6. Setelah itu, *IMS Core* melanjutkan pesan yang dikirim oleh *Application Server* tadi ke *client* untuk memberitahu bahwa layanan VoD telah siap untuk digunakan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

7. *Application Server* mengirimkan pesan ke *IMS Core* dengan status: 200 OK, yang menandakan bahwa layanan VoD telah dijalankan dan telah terhubung dengan alamat rtsp://127.0.0.1:8000/channel2.
8. *IMS Core* melanjutkan pesan yang telah dikirimkan oleh *Application Server* tadi ke *client.*
9. Kemudian *client* mengirimkan pesan ACK ke *IMS Core*, yang menandakan bahwa *client* telah menerima *response* final dari *request INVITE* yang telah dikirimkannya.
10. *IMS Core* kemudian melanjutkan pesan ACK tersebut ke *Application Server.*

4.1.3 *Message Flow* pada Saat Pemutusan Layanan *Video on Demand*

Uji coba ini dilakukan dengan pemutusan layanan *Video on Demand (VoD)* yang telah digunakan oleh *client*, dengan alamat sip:channel2@open-ims.test, sehingga diperoleh *message flow* seperti berikut:

Tabel 4.3 *Message flow* pada saat pemutusan layanan VoD

No.	Source	Destination	Protocol	Info
1.	*Client*	*IMS Core*	SIP	Request: BYE sip:channel2@iptv.open-ims.test
2.	*IMS Core*	*Application Server*	SIP	Request: BYE sip:channel2@iptv.open-ims.test
3.	*Application Server*	*IMS Core*	SIP	Status: 200 OK
4.	*IMS Core*	*Client*	SIP	Status: 200 OK

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Berdasarkan Tabel 4.3, dapat dianalisis bahwa *message flow* untuk proses pemutusan layanan VoD adalah sebagai berikut:

1. *Client* mengirimkan *request BYE* ke *IMS Core* untuk mengakhiri penggunaan layanan VoD yang telah diminta.
2. *IMS Core* melanjutkan pesan tersebut ke *Application Server.*
3. *Application Server* membalas pesan tersebut ke *IMS Core* dengan status: 200 OK, yang menandakan bahwa penggunaan layanan VoD sudah diakhiri.
4. Kemudian *IMS Core* melanjutkan pesan yang telah dikirimkan oleh *Application Server* tersebut ke *client.*

4.1.4 *Message Flow* pada Saat Deregistrasi

Uji coba ini dilakukan dengan melakukan proses deregistrasi *client* dengan IMPU sip:alice@open-ims.test, sehingga diperoleh *message flow* seperti berikut:

Tabel 4.4 *Message flow* pada saat 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 (0 bindings)
3.	*Client*	*IMS Core*	SIP	Request: REGISTER sip:open-ims.test
4.	*IMS Core*	*Client*	SIP	Status: 200 OK - SAR successful and registrar saved (1 bindings)
5.	*IMS Core*	*Client*	SIP	Request: NOTIFY sip:alice@10.0.2.15:5061
6.	*Client*	*IMS Core*	SIP	Status: 200 OK

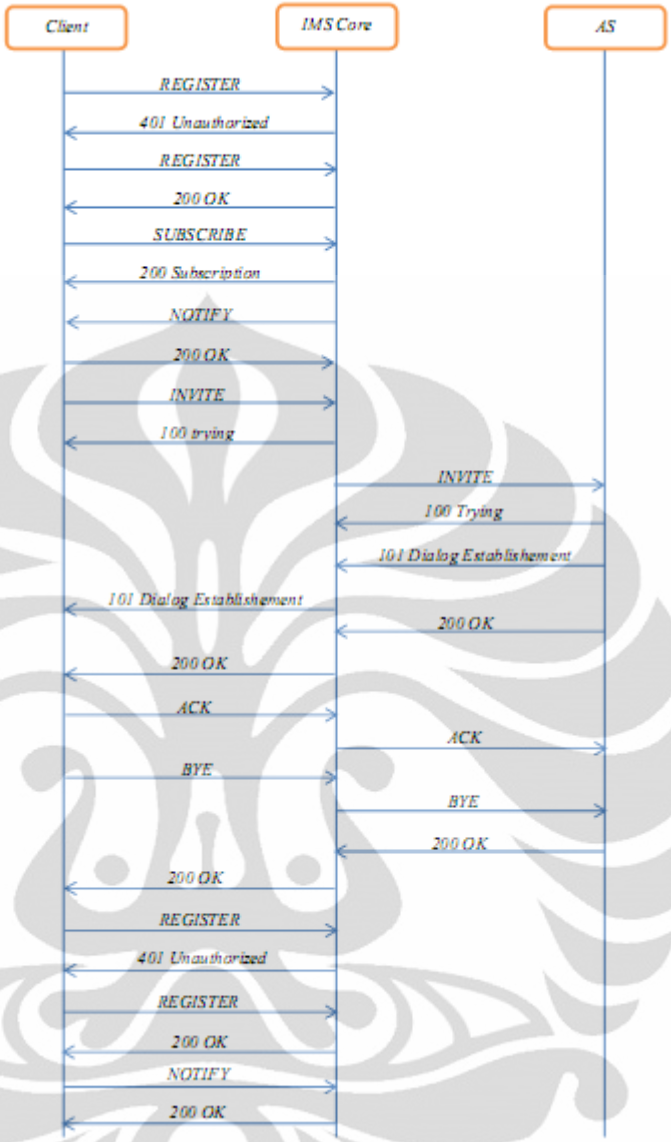
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Berdasarkan Tabel 4.4, terlihat bahwa *message flow* untuk proses deregistrasi *client* hampir sama dengan *message flow* untuk proses registrasi *client.* Hal ini dikarenakan protokol SIP tidak mengenal *request message DEREGISTER.* Oleh karena itu, *request message* yang digunakan pada saat registrasi dan deregistrasi adalah *request message REGISTER.* Meskipun sama-sama menggunakan *request message REGISTER,* ada beberapa hal yang membedakan antara *message flow* untuk proses registrasi dan deregistrasi, yaitu:

a. Pada proses deregistrasi, tidak terdapat tahap *SUBSCRIBE.* Sedangkan pada proses registrasi terdapat tahap *SUBSCRIBE.*
b. Apabila dilihat dengan menggunakan wireshark, besarnya nilai "*Expires*" pada saat proses registrasi adalah 600000. Sedangkan pada saat proses deregistrasi, "*Expires*" bernilai 0.

Secara keseluruhan, *message flow* mulai dari saat registrasi sampai dengan deregistrasi adalah sebagai berikut:

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

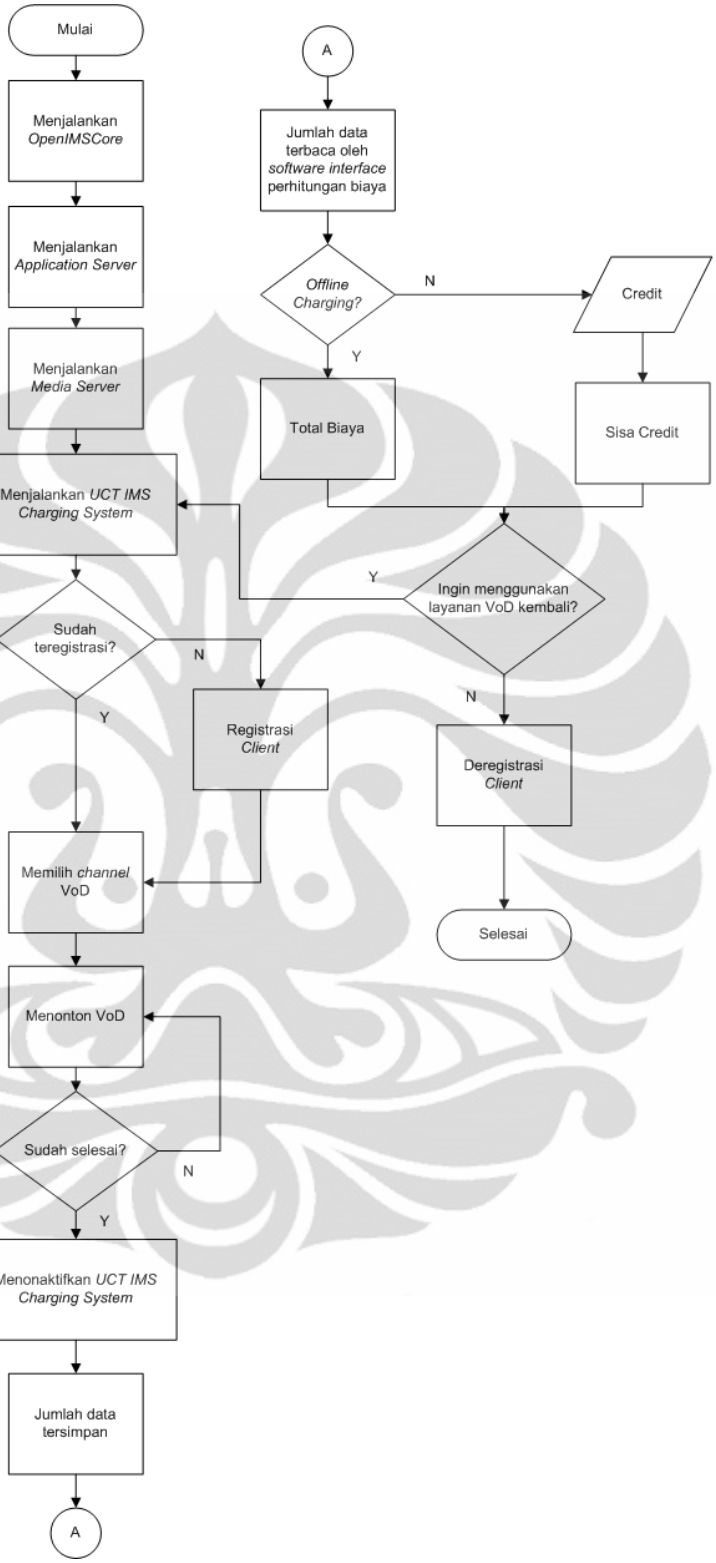


Gambar 4.1 *Message flow* registrasi - deregistrasi

4.2 Analisis Implementasi *UCT IMS Charging System* untuk Layanan Video on Demand

Pada uji coba kali ini, digunakan *UCT IMS Charging System* yang dikembangkan oleh University of Cape Town. Sistem *charging* ini dapat digunakan untuk *offline charging* maupun *online charging.* Adapun *flow chart* untuk uji coba ini adalah sebagai berikut:

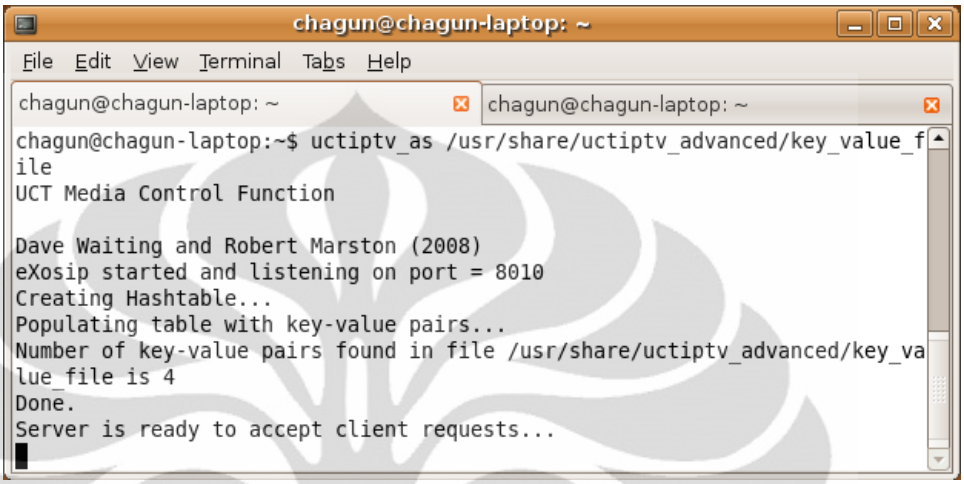
Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 4.2 *Flow Chart*

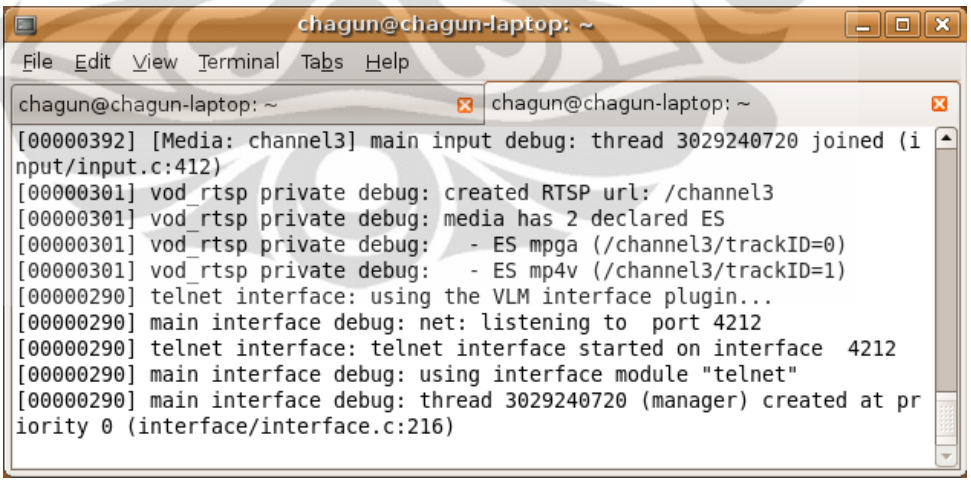
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Sesuai dengan Gambar 4.2, tahap pertama yang harus dilakukan adalah menjalankan *OpenIMSCore* terlebih dahulu, kemudian dilanjutkan dengan *application server* dan *media server.*



Gambar 4.3 *Application Server* telah aktif

Gambar 4.3 menunjukkan bahwa *application server* telah aktif dan siap untuk menerima *request* dari *client.*

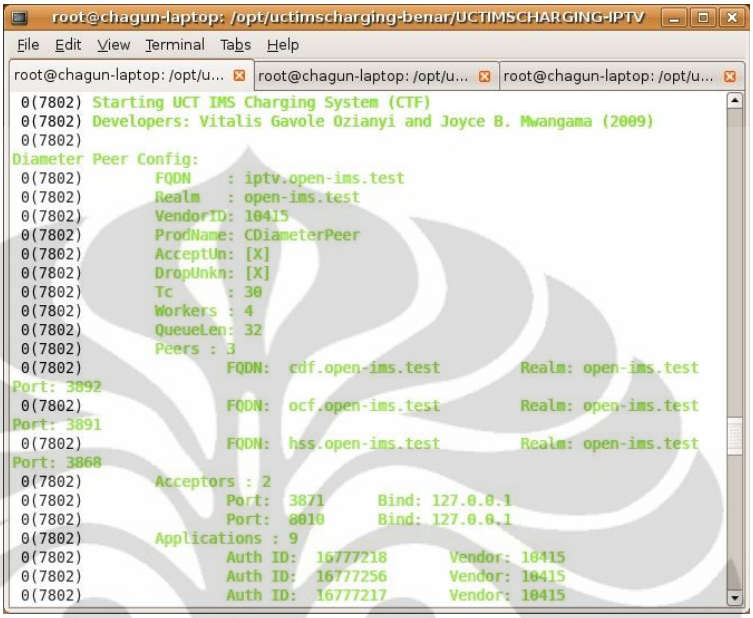


Gambar 4.4 *Media Server* telah aktif

Sedangkan Gambar 4.4, menunjukkan bahwa *media server* telah aktif dan siap untuk digunakan. Setelah itu, barulah menjalankan *UCT IMS Charging System.*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Pertama-tama, jalankan *Charging Trigger Function* seperti pada Gambar 4.5 berikut:

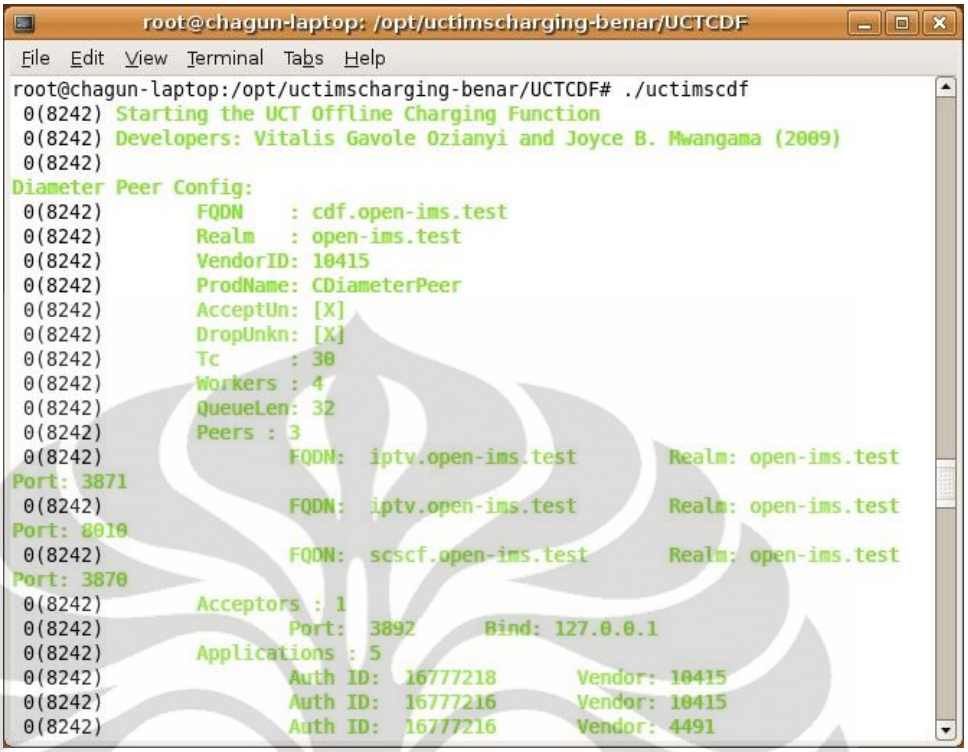


Gambar 4.5 CTF telah aktif

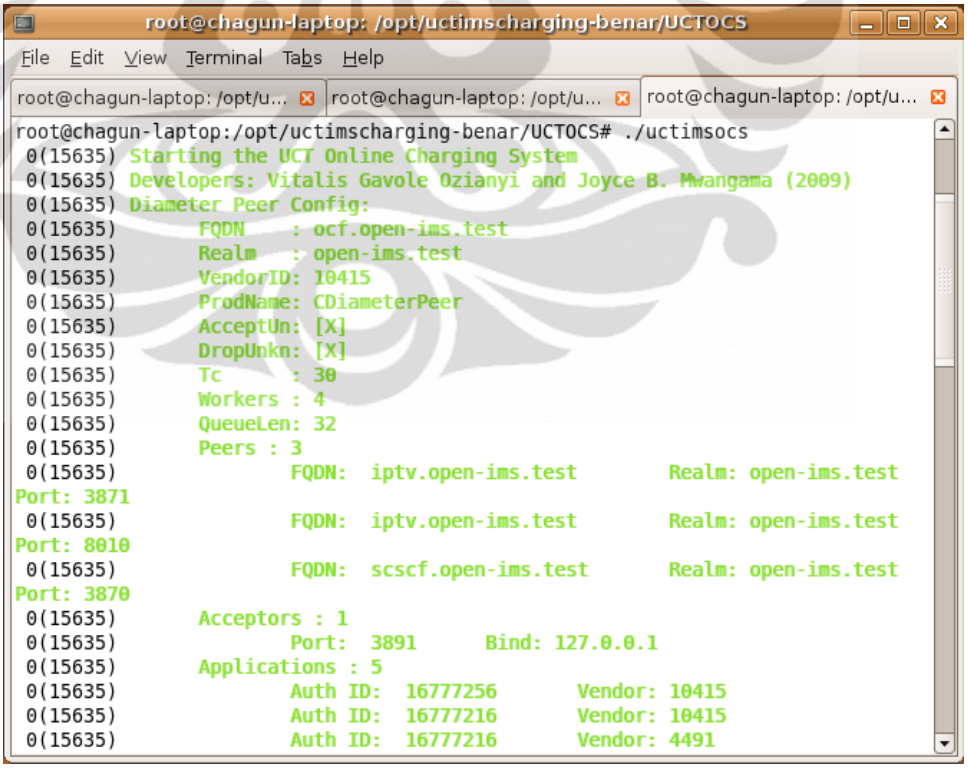
Gambar 4.5 menandakan bahwa *Charging Trigger Function (CTF)* telah aktif. Dari gambar tersebut, telihat bahwa CTF telah nantinya akan terhubung dengan *Charging Data Function (CDF)*, *Online Charging Function (OCF)*, dan HSS. Akan tetapi, hal tersebut dapat terjadi apabila CDF, OCF, dan HSS telah aktif. Dari Gambar 4.5 juga terlihat bahwa ada beberapa *interface* yang digunakan. Hal tersebut ditunjukkan dengan *"Auth ID"* yang terlihat.

Langkah selanjutnya adalah mengaktifkan CDF untuk *offline charging* atau OCF untuk *online charging.* Apabila mengaktifkan CDF, maka akan terlihat seperti pada Gambar 4.6. Sedangkan apabila mengaktifkan OCF, maka akan terlihat seperti pada Gambar 4.7. Sama halnya dengan CTF yang terhubung dengan komponen lainnya, CDF maupun OCF juga nantinya akan terhubung dengan komponen lainnya.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



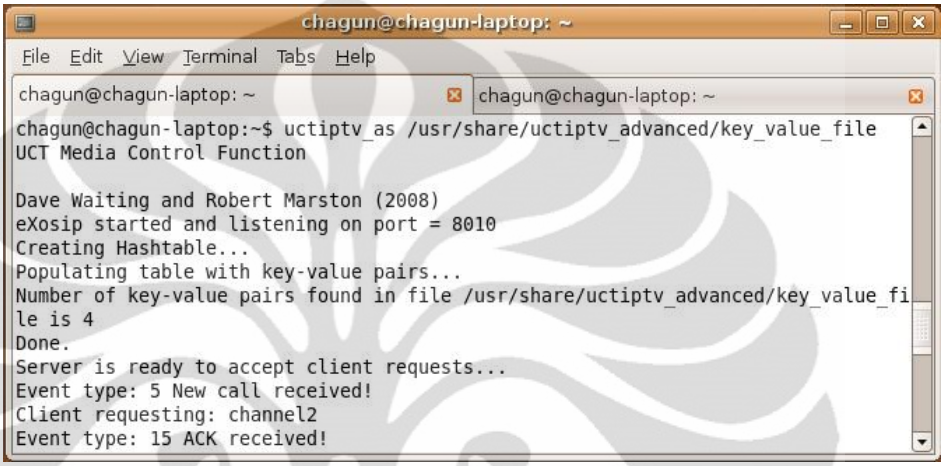
Gambar 4.6 CDF telah aktif



Gambar 4.7 OCF telah aktif

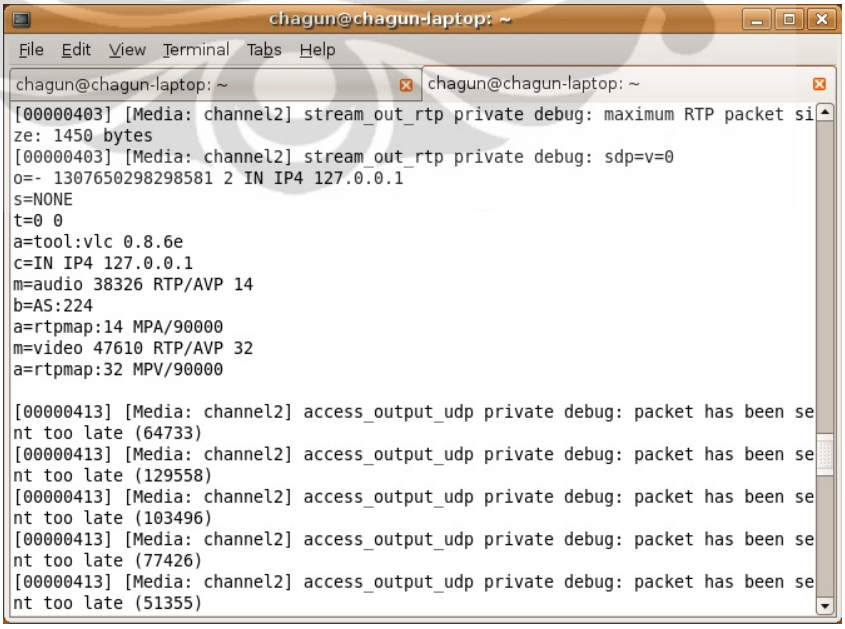
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Setelah semuanya telah aktif, maka tahap selanjutnya *client* melakukan registrasi dan meminta layanan *Video on Demand (VoD)* dengan memilih *channel* yang ada. Caranya telah dijelaskan pada bab sebelumnya. Setelah VoD dijalankan, maka akan terjadi perubahan-perubahan pada *application server, media server,* CTF, dan CDF/OCF.



Gambar 4.8 *Application Server* setelah VoD dijalankan

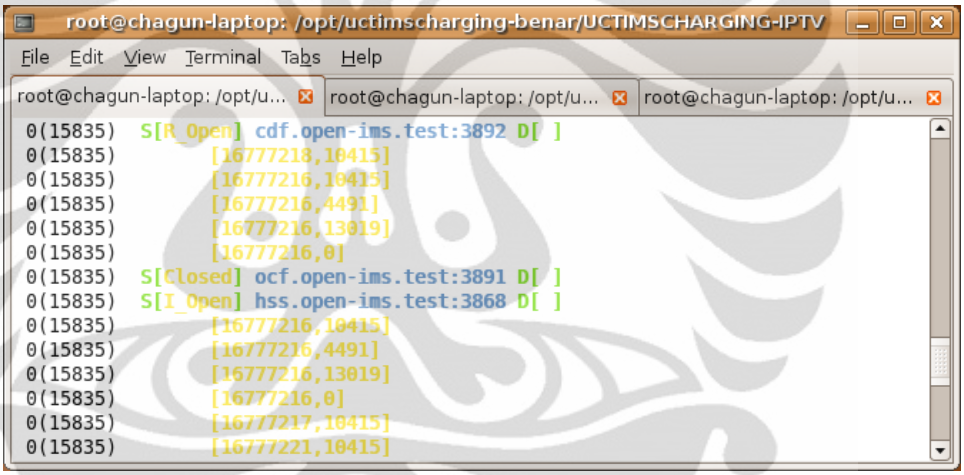
Berdasarkan Gambar 4.8, terlihat bahwa *client* telah me-*request* VoD pada *channel 2* dan VoD pun telah diterima oleh *client.*



Gambar 4.9 *Media Server* setelah VoD dijalankan

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

Gambar 4.9 merupakan tampilan *media server* ketika VoD sedang dijalankan. Pada gambar tersebut terlihat adanya SDP (*Session Description Protocol*), yang ditandai dengan "o=", "s=", "t=", "a=", "c=", "m=", dan "b=". Untuk "o=" menandakan *owner/creator* dan *session identifier*, "s=" menandakan *session name,* "t=" menandakan *time the session* telah aktif, "a" membawa informasi mengenai media yang digunakan dan menandakan pemetaan dari nomor RTP *payload*, "c=" menunjukkan informasi mengenai koneksi, "m=" membawa informasi mengenai *media type* dan *transport protocol* yang digunakan, "b=" menyampaikan informasi mengenai *bandwidth.* Selain itu, Gambar 4.9 juga memperlihatkan proses pengiriman *packet* ke *client.*



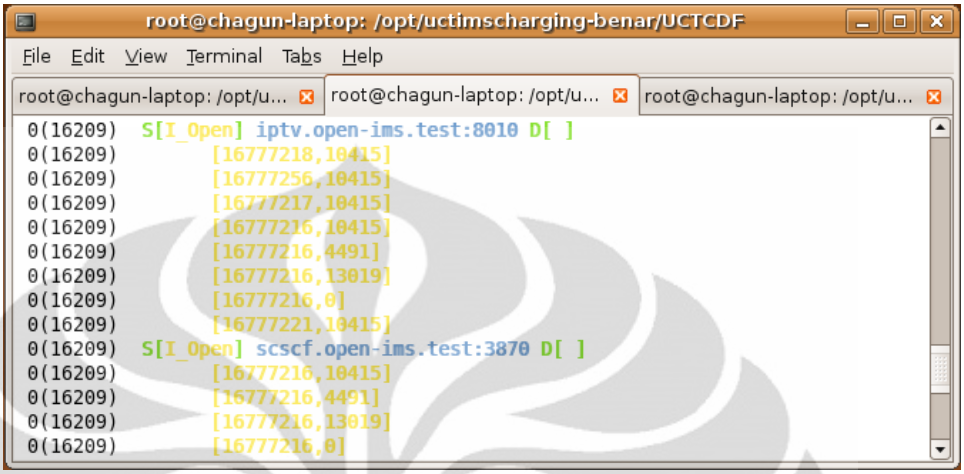
Gambar 4.10 Tampilan CTF setelah VoD dijalankan

Gambar 4.10 merupakan tampilan CTF ketika VoD sedang dijalankan. Pada gambar tersebut terlihat bahwa telah terjadi hubungan antara CTF dengan CDF dan HSS. Hal ini ditandai dengan status "*open*" pada masing-masing komponen. [16777218] merupakan *interface* Rf yang menghubungkan antara CTF dengan CDF, sedangkan [10415] merupakan vendornya. Begitu juga untuk [16777216, 10415], yang merupakan sebuah *interface* dan vendor yang menghubungkan antara CTF dengan HSS.

Sedangkan untuk tampilan CDF (*offline charging*) ketika VoD sedang dijalankan, terlihat pada Gambar 4.11. Pada prinsipnya sama dengan yang terjadi pada CTF, yang membedakan hanyalah komponen-

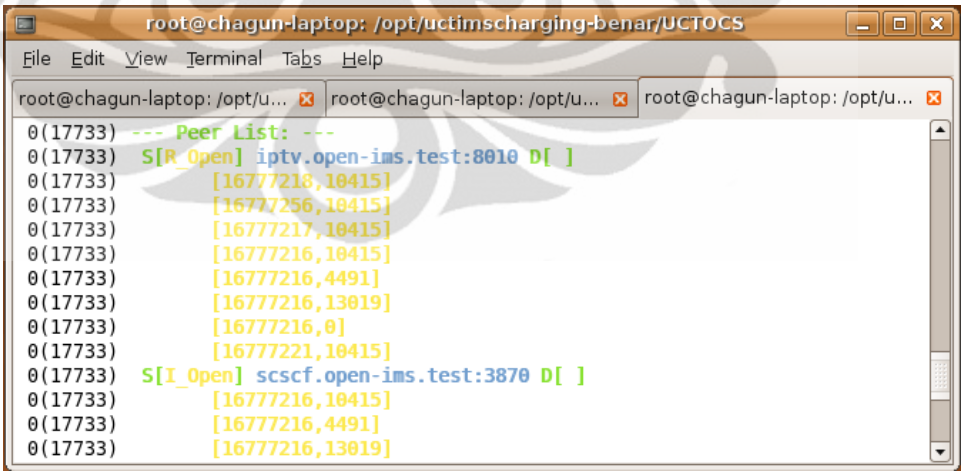
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

komponen yang terhubung dan *interface* serta vendor yang menghubungkannya.



Gambar 4.11 Tampilan CDF setelah VoD dijalankan

Untuk *online charging*, komponen yang terhubung dengan CTF adalah OCF. CTF dan OCF terhubung dengan *interface* Ro, yang pada Gambar 4.12 ditandai dengan [16777256]. Selain terhubung dengan CTF, OCF juga terhubung dengan S-CSCF sama halnya dengan CDF.

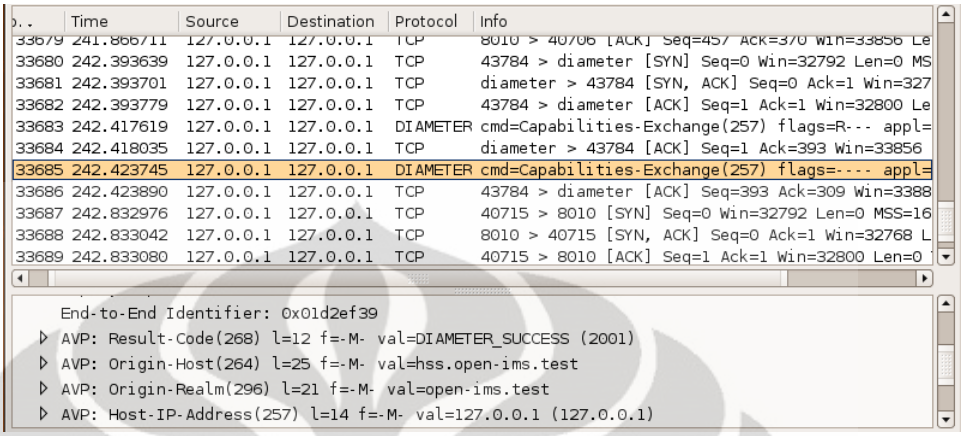


Gambar 4.12 Tampilan OCF setelah VoD dijalankan

Ketika *charging system* telah dijalankan, *application server* akan menghubungi HSS untuk meminta data-data tentang *client.* Pada Gambar

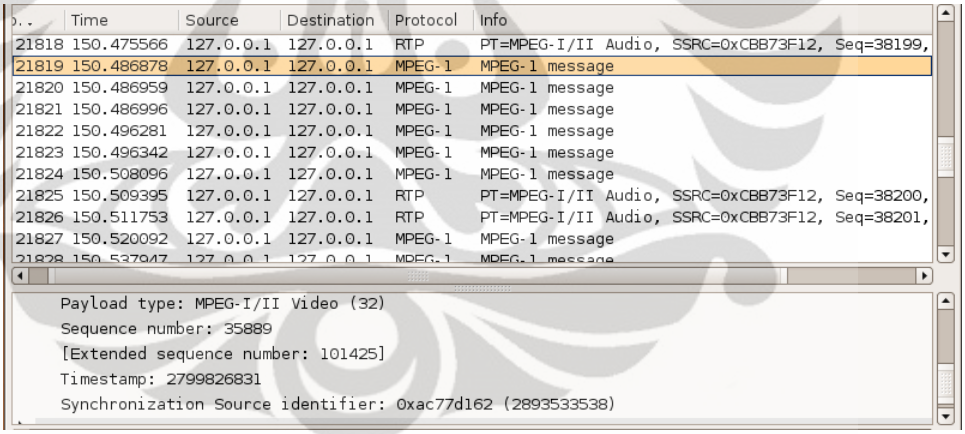
Perancangan dan ..., Chandra Gunawan, FT UI, 2011

4.13, terlihat bahwa antara HSS dengan *application server* telah berhasil terkoneksi.



Gambar 4.13 Tampilan wireshark HSS – IPTV

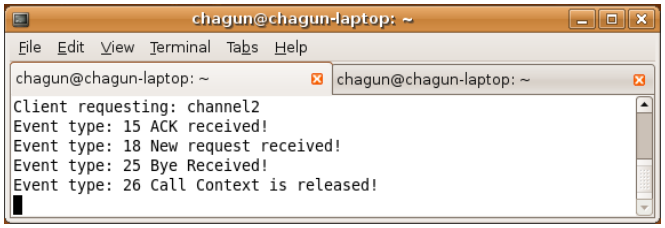
Berikut ini adalah tampilan pada wireshark mengenai proses pengiriman *packet-packet* pada saat proses *charging* sedang berlangsung.



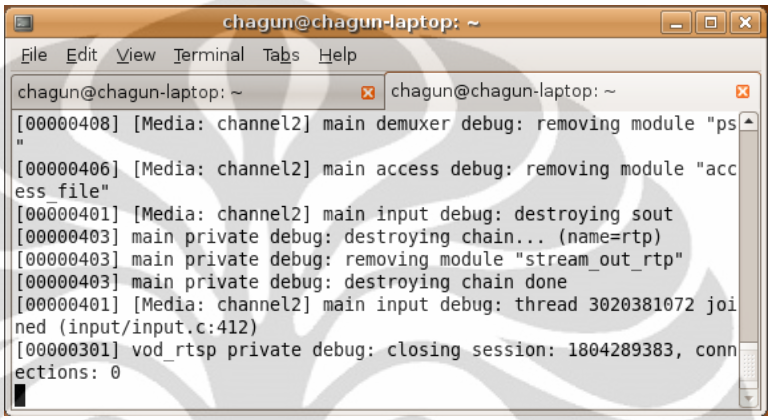
Gambar 4.14 Proses pengiriman *packet* pada wireshark

Ketika *client* telah selesai menggunakan layanan *Video on Demand (VoD)*, maka *application server* dan *media server* akan dalam berada pada kondisi *standby*. Maksudnya adalah *server* tersebut siap untuk digunakan ketika ada *client* yang ingin menggunakannya. Sedangkan *IMS Core* akan berjalan normal seperti biasanya.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



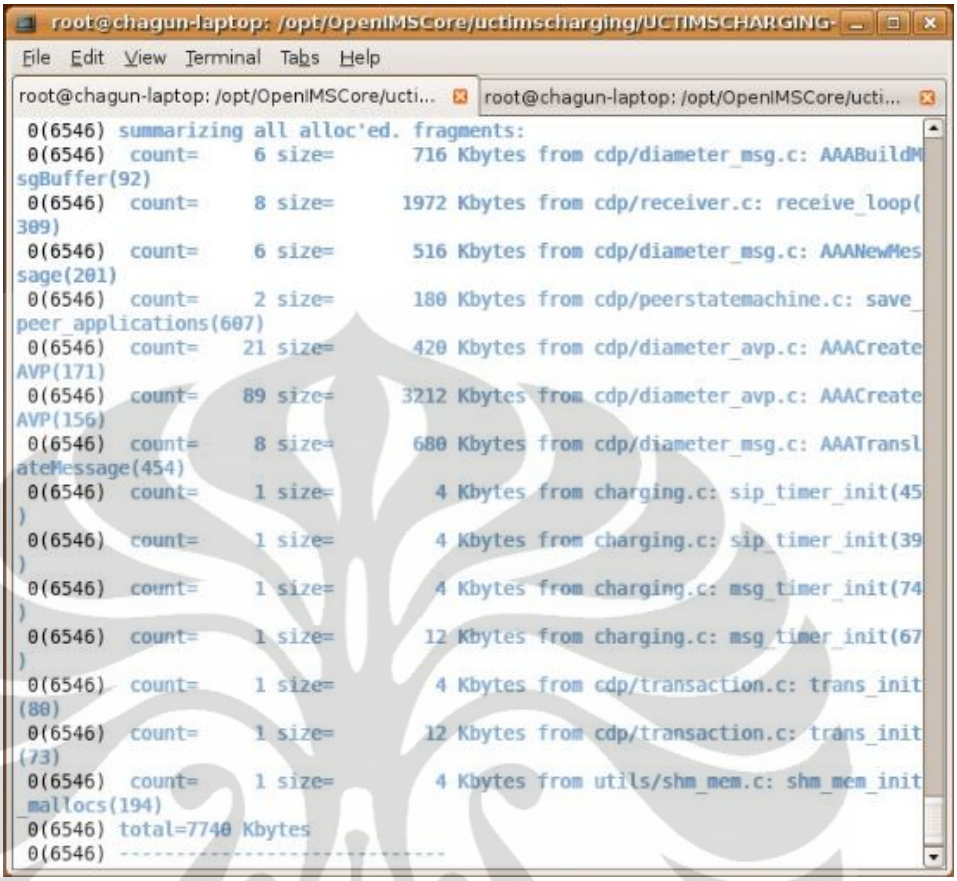
Gambar 4.15 Tampilan *application server* pada kondisi *standby*



Gambar 4.16 Tampilan *media server* pada kondisi *standby*

Lain halnya dengan *charging system*, *charging system* harus dinonaktifkan untuk melihat banyaknya jumlah *bytes* yang telah digunakan selama proses implementasi *UCT IMS Charging System* pada layanan *Video on Demand (VoD)* berlangsung. Berikut ini adalah tampilan ketika *charging system* dinonaktifkan.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

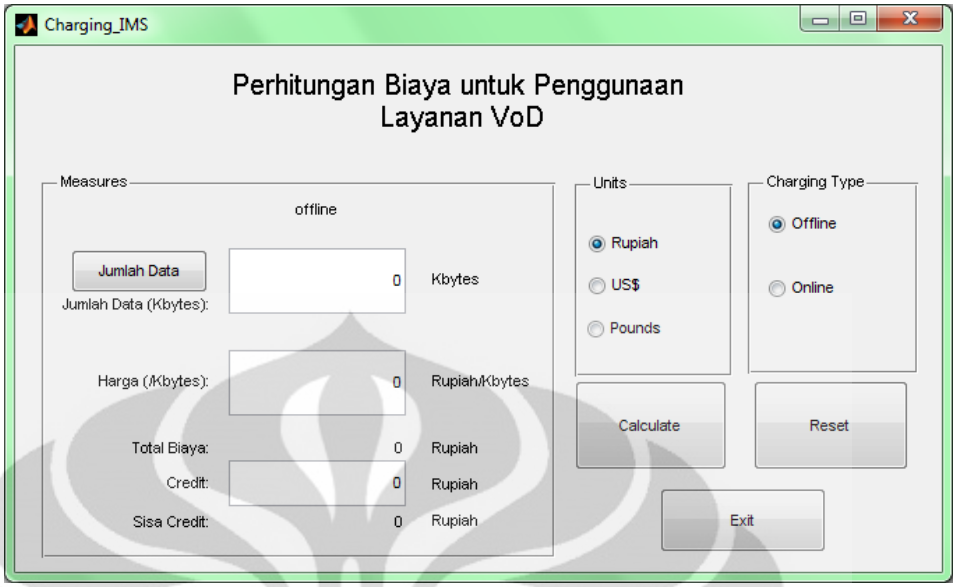


Gambar 4.17 Tampilan CTF setelah *charging system* dinonaktifkan

Dari Gambar 4.17, terlihat bahwa terjadi perhitungan jumlah *bytes* yang digunakan selama *charging system* sedang diaktifkan. Jumlah *bytes* tersebutlah yang kemudian akan digunakan untuk perhitungan biaya yang harus dikeluarkan oleh *client* setelah menggunakan layanan *Video on Demand (VoD).*

Setelah mendapatkan jumlah *bytes* dari proses *charging,* kemudian jumlah *bytes* tersebut akan disimpan ke dalam suatu file dengan nama "jumlah.dat". Setelah itu, file tersebut akan digunakan sebagai masukan dalam *software interface* perhitungan biaya. Dengan demikian, jumlah data dari proses *charging* akan secara otomatis terisi ke dalam *software interface* tersebut. Adapun bentuk *software interface* tersebut seperti Gambar 4.18.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 4.18 Tampilan *software interface* perhitungan biaya

keterangan:

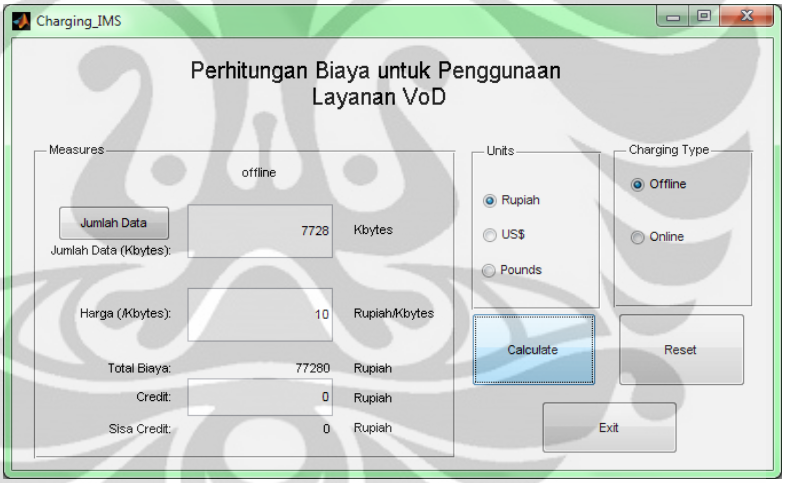
- ***Calculate***: untuk menghitung besarnya biaya.
- ***Reset***: untuk mengembalikan ke kondisi awal.
- ***Exit***: untuk keluar dari *software interface* perhitungan biaya.
- **Jumlah Data (*Kbytes*)**: merupakan masukan yang diperoleh dari proses *charging.* Masukan ini hanya bisa diisi oleh angka dan merupakan bilangan genap. Selain itu, akan timbul pesan *error*.
- **Harga (*Kbytes*):** merupakan masukan yang berisi harga per *Kbytes.*
- **Total Biaya**: merupakan keluaran untuk proses perhitungan ini.
- ***Credit:*** seperti layaknya pulsa pada telepon selular atau *handphone.* Digunakan pada *online charging.*
- **Sisa *Credit:*** merupakan sisa dari penggunaan *credit* sebelumnya.
- **Rupiah**: merupakan satuan mata uang rupiah.
- **US$**: merupakan satuan mata uang dollar US, dimana 1 US$ sama dengan 10.000 rupiah.

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

- **Pounds**: merupakan satuan mata uang poundsterling, dimana 1 poundsterling sama dengan 15.000 rupiah.
- ***Online:*** merupakan *online charging.*
- ***Offline:*** merupakan *offline charging.*

Contoh:

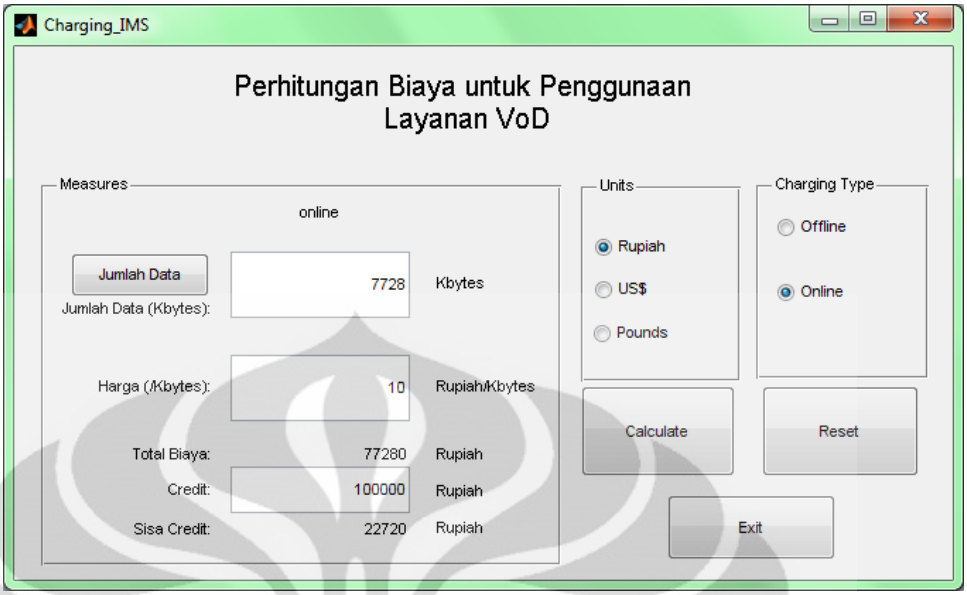
Alice telah menggunakan layanan VoD dengan menggunakan *offline charging*. Setelah dilihat, ternyata jumlah *bytes* yang diperoleh dari proses *charging* tersebut adalah 7.728 *Kbytes.* Apabila harga per *Kbytes* sebesar 10 rupiah, maka Alice dikenakan biaya sebesar 77.280 rupiah atas penggunaan layanan VoD tersebut.



Gambar 4.19 Perhitungan biaya untuk *offline charging*

Gambar 4.19 memperlihatkan total biaya yang dikenakan kepada Alice atas penggunaan layanan VoD secara *offline charging*. Selain untuk *offline charging*, penggunaan layanan VoD juga bisa digunakan untuk *online charging.*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011



Gambar 4.20 Perhitungan biaya untuk *online charging*

Dari Gambar 4.20, terlihat bahwa proses perhitungan biaya dilakukan untuk *online charging*. Pada kasus ini, diasumsikan *client* memiliki jumlah *credit* sebesar 100.000 rupiah. Karena biaya yang dikenakan atas penggunaan layanan VoD sebesar 77.280 rupiah, maka *credit* yang tersisa sebesar 22.720 rupiah. Hal ini sesuai dengan prinsip kerja dari *online charging*, dimana terjadi pengurangan besarnya *credit* yang dimiliki oleh *client.*

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

BAB 5
KESIMPULAN

Berdasarkan pembahasan di dalam skripsi ini, dapat disimpulkan beberapa hal, yaitu:

1. *OpenIMSCore* ditambah dengan adanya *Application Server, Media Server*, *UCT IMS Charging System,* dan *software interface* perhitungan biaya, dapat digunakan untuk implementasi sistem *charging* layanan *Video on Demand* (VoD).
2. *Message flow* penggunaan layanan IPTV (VoD) pada IMS, mulai dari proses registrasi, meminta layanan VoD, pemutusan layanan VoD, sampai dengan deregistrasi terlihat seperti pada Gambar 4.1.
3. Ada tiga elemen utama pada *UCT IMS Charging System* yang berperan penting dalam proses *charging*, yaitu *Charging Trigger Function* (CTF), *Charging Data Function* (CDF), dan *Online Charging Function* (OCF).
4. Proses *charging* pada IMS dapat diimplementasikan secara *online* maupun *offline.*
5. Setelah memperoleh data dari *UCT IMS Charging System,* data tersebut akan secara otomatis masuk ke dalam *software interface* perhitungan biaya untuk dilakukan penghitungan total biaya yang dikenakan kepada *client* atas penggunaan layanan IPTV (VoD).

Perancangan dan ..., Chandra Gunawan, FT UI, 2011

DAFTAR REFERENSI

[1] Ozianyi, V. G., & Ventura, N. 2009. *Charging in IP Multimedia Networks.* University of Cape Town.

[2] Menyongsong Era Konvergensi Telekomunikasi dan Informasi. Diakses 15 Desember 2010, dari myanuar.wordpress.com http://myanuar.wordpress.com/2008/12/31/menyongsong-era-konvergensi-telekomunikasi-dan-informasi/

[3] M. Poikselka, G. Mayer. 2009. *The IMS: IP Multimedia Concepts and Services, Third edition* (hal 5-7). John Wiley & Sons, UK.

[4] S. A. Ahson, M. Ilyas. 2009. *IP Multimedia Subsystem (IMS) Handbook* (hal 29-32). CRC Press Taylor & Francis Group, USA.

[5] *Video on demand.* Diakses 10 Juni 2011, dari wikipedia.org http://id.wikipedia.org/wiki/Video_on_demand

[6] Layanan Multimedia *Streaming*. Diakses 10 Juni 2011, dari wenythepooh.wordpress.com https://wenythepooh.wordpress.com/2010/10/13/layanan-multimedia-streaming/

[7] SIP (*Session Initiation Protocol*). Diakses 11 Juni 2011, dari leak.web.id http://www.leak.web.id/2009/04/sip-session-initiation-protocol.html

[8] *Session Initiation Protocol.* Diakses 11 Juni 2011, dari wikipedia.org http://en.wikipedia.org/wiki/Session_Initiation_Protocol

[9] *How to Setup UCT IMS IPTV Charging System.* Diakses 30 Mei 2011, dari uctimsclient.berlios.de http://uctimsclient.berlios.de/uctimscharging.html

[10] *OpenIMSCore.* Diakses 30 Mei 2011, dari openimscore.org http://www.openimscore.org/

[11] *How to Setup UCT Advanced IPtv.* Diakses 30 Mei 2011, dari uctimsclient.berlios.de http://uctimsclient.berlios.de/uctiptv_advanced_howto.html