



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI
LAYANAN INSTANT MESSAGING
PADA IMPLEMENTASI OPEN IMS CORE**

SKRIPSI

**MARK SWAPO NOVEANTO
0606074104**

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



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI
LAYANAN INSTANT MESSAGING
PADA IMPLEMENTASI OPEN IMS CORE**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**MARK SWAPO NOVEANTO
0606074104**

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

i

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Mark Swapo Noveanto

NPM : 0606074104

Tanda Tangan : 

Tanggal : 30 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Mark Swapo Noveanto
NPM : 0606074104
Program Studi : Teknik Elektro
Judul Skripsi : Analisa Performansi Layanan Instant Messaging
pada Implementasi 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

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas kehendak dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi dilakukan sebagai salah satu syarat untuk menjadi Sarjana Teknik di Departemen Teknik Elektro FTUI. Saya menyadari bahwa tanpa bantuan banyak pihak, skripsi ini tidak mungkin terselesaikan. Oleh sebab itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada pihak-pihak yang membantu saya dalam segala hal mengenai penyusunan skripsi ini yaitu:

- 1) Ir. Djamhari Sirat M.Sc, Ph.D selaku dosen Pembimbing Akademik dan Pembimbing seminar dan skripsi saya yang selalu memberikan visi yang memukau dan motivasi yang hampir tak terhingga.
- 2) Pak Noor Iza (Depkominfo), Pak Yulianus (Indosat), Pak Bambang Soekanto (Telkom), dan Pak Panca (Telkomsel) yang berhasil memberikan arahan secara umum dan teknis terhadap kebutuhan dunia telekomunikasi sekarang dan di masa yang akan datang.
- 3) Rekan-rekan satu bimbingan: R. Kharisma, Deolens Manurung, Hartono Hung, dan Zaneta Pelangi yang jatuh bangun berjuang untuk membangun jaringan implementasi Open IMS Core bersama-sama.
- 4) Kedua orang tua saya yang dengan sabar membiayai seluruh kehidupan saya sampai berhasil menyelesaikan kuliah di Universitas Indonesia. Tak lupa, adik dan kakak tercinta Rossa dan Reno yang selalu mendoakan saya.
- 5) Seluruh pihak di Teknik Elektro FTUI dan teman-teman akrab di Kelapa Gading yang tidak dapat saya sebutkan satu per satu.

Depok, 30 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 : Mark Swapo Noveanto

NPM : 0606074104

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 PERFORMANSI LAYANAN INSTANT MESSAGING
PADA IMPLEMENTASI 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 : 30 Juni 2010

Yang Menyatakan



(Mark Swapo Noveanto)

ABSTRAK

Nama : Mark Swapo Noveanto
Program Studi : Teknik Elektro
Judul : ANALISA PERFORMANSI LAYANAN INSTANT
MESSAGING PADA IMPLEMENTASI OPEN IMS CORE

Skripsi ini membahas tentang performansi layanan Instant Messaging pada jaringan implementasi Open IMS Core. Jaringan ini menggunakan sebuah IMS Core Server, dua router, sebuah switch dan dua laptop yang digunakan sebagai client. Dari kedua router, bandwidth dikontrol sehingga besar bandwidth pada hubungan keduanya dapat dijaga.

Perubahan bandwidth menggunakan standar ITU-T seri V, IDSL, dan HDSL. Pada setiap bandwidth, parameter QoS *delay* diukur sebanyak tiga kali percobaan melalui SIP MESSAGE yang diperoleh. Hasil *delay* ketiga percobaan dirata-rata untuk melihat nilai *delay* pada bandwidth tersebut. Standar minimum fungsi IM pada V.32 (9,6 kbps) sedangkan untuk performansi yang optimal didapatkan pada standar V.92 (56 kbps).

Kata Kunci:

Instant Messaging, Open IMS Core, *bandwidth*, *delay*

ABSTRACT

Name : Mark Swapo Noveanto
Study Program : Teknik Elektro
Title : PERFORMANCE ANALYSIS INSTANT MESSAGING SERVICE IN OPEN IMS CORE IMPLEMENTATION

This final assignment discusses about Instant Messaging service performance in Open IMS Core implementation network. This network is using an IMS Core Server, two router, a switch, and two laptop as clients. From two router, the bandwidth is controlled so the volume of the bandwidth can be maintained.

ITU-T V Series, IDSL, and HDSL are used as the standard to change the bandwidth. In each bandwidth, *delay* QoS parameter is measured by three times experiments from SIP MESSAGE that obtainable. The *delay* result is evenly divided to get the *delay* mean score for this bandwidth. For IM function, the minimum standard is V.32 (9,6 kbps). Furthermore, for optimum performance, IM can use V.92 (56 kbps) standard.

Keywords:

Instant Messaging, Open IMS Core, *bandwidth*, *delay*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
1. PENDAHULUAN	1
1.1 Latar Belakang Skripsi	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 Session Initiation Protocol (SIP)	6
2.1.1 Komponen-komponen SIP	6
2.1.2 SIP Request/Method dan Response	7
2.1.2.1 Request/Method	7
2.1.2.2 Response	8
2.1.3 Session Description Protocol (SDP)	8
2.1.4 SIP Pada Instant Messaging	9
2.2 IP Multimedia Subsystem (IMS)	11
2.2.1 Home Subscriber Server (HSS)	12
2.2.2 Proxy Call Session Control Function (P-CSCF)	13
2.2.3 Interrogating Call Session Control Function (I-CSCF)	13
2.2.4 Serving Call Session Control Function (S-CSCF)	14
2.3 Layanan Instant Messaging	14
2.4 Parameter QoS Instant Messaging	16
3. INSTALASI IMS DAN PROSES INSTANT MESSAGING	18
3.1 Open IMS Core	18
3.1.1 Instalasi Open IMS Core	19
3.1.2 Konfigurasi DHCP dan DNS	21
3.1.3 Pengaktifkan Perangkat Open IMS Core	22

3.1.4 Testing Open IMS Core.....	23
3.2 IMS Client	24
3.2.1 MyMonster.....	24
3.2.2 UCT IMS Client	25
3.3 Skenario Jaringan Penelitian.....	26
3.4 Proses Registrasi	27
3.4.1 Pendaftaran pelanggan melalui HSS	27
3.4.2 Registrasi menggunakan client	29
3.5 Inisiasi Percakapan (Chatting)	30
4. PENGUKURAN DAN ANALISA PARAMETER QoS.....	32
4.1 Prosedur dan Variabel	32
4.2 Pengambilan dan Pengolahan Data	33
4.2.1 V.22	33
4.2.2 V.32	34
4.2.3 V.34	35
4.2.4 V.92	37
4.2.5 IDSL	38
4.2.6 HDSL.....	40
4.2.7 Message Flow.....	41
4.3 Analisa <i>Delay</i>	42
4.3 Analisa Tambahan.....	45
5. KESIMPULAN.....	46
DAFTAR ACUAN	47
LAMPIRAN	48

DAFTAR TABEL

Tabel 2.1 Parameter QoS Jaringan	17
Tabel 4.1 Data I V.32	34
Tabel 4.2 Data II V.32	34
Tabel 4.3 Data III V.32	35
Tabel 4.4 Data I V.34	35
Tabel 4.5 Data II V.34	36
Tabel 4.6 Data III V.34	36
Tabel 4.7 Data I V.92	37
Tabel 4.8 Data II V.92	37
Tabel 4.9 Data III V.92	38
Tabel 4.10 Data I IDSL	38
Tabel 4.11 Data II IDSL	39
Tabel 4.12 Data III IDSL	39
Tabel 4.13 Data I HDSL	40
Tabel 4.14 Data II HDSL	40
Tabel 4.15 Data III HDSL	41

DAFTAR GAMBAR

Gambar 2.1 Open IMS Core pada Open IMS Playground	5
Gambar 2.2 OSI Layer	9
Gambar 2.3 Bagan Paket SIP MESSAGE	11
Gambar 2.4 Elemen-elemen IMS	11
Gambar 2.5 Core IMS	12
Gambar 2.6 CSCFs	13
Gambar 3.1 Open IMS Core	19
Gambar 3.2 Halaman Otentikasi FhoSS	23
Gambar 3.3 Logo MyMonster	24
Gambar 3.4 Tampilan Awal MyMonster	25
Gambar 3.5 UCT Badge	25
Gambar 3.6 Tampilan Awal UCT IMS Client	26
Gambar 3.7 Skenario Jaringan Penelitian	26
Gambar 3.8 Registrasi IMSU	28
Gambar 3.9 Registrasi IMPI	29
Gambar 3.10 Registrasi IMPU	29
Gambar 3.11 Preferences pada UCT IMS Client	30
Gambar 3.12 Inisiasi Panggilan dengan Public Identity	31
Gambar 3.13 Dialog Box untuk Proses <i>Chatting</i>	31
Gambar 4.1 Message Flow Registrasi	41
Gambar 4.2 Message Flow Chatting	42
Gambar 4.3 Grafik Hubungan <i>Bandwidth</i> dengan <i>Delay</i>	43

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Upaya untuk menekan biaya kapital dan operasional menjadi perhatian di Indonesia karena muncul budaya “perang tarif” yang saling menghancurkan operator seluler maupun operator fixed. Oleh sebab itu, dibutuhkan suatu strategi baru dimana jaringan *IP Multimedia Subsystem* (IMS) diharapkan menjadi ujung tombaknya. Strategi pemanfaatan jaringan IMS diharapkan mampu meningkatkan kualitas layanan berbasis IP bukan hanya sekedar “best effort” saja.

Kemampuan IMS dalam mengontrol kualitas jaringan patut diuji dengan aplikasi-aplikasi yang memiliki potensi penghasilan tinggi atau yang dikenal dengan istilah “killer application”. Dengan demikian, tujuan untuk peningkatan kualitas layanan telekomunikasi ini juga diimbangi dengan pemasukan yang baik untuk pihak operator.

Instant Messaging sebagai salah satu layanan yang memiliki potensi penghasilan tinggi di dunia telekomunikasi dapat dimasukkan ke dalam kategori “killer application”. Sehingga, perlu dilakukan emulasi jaringan IMS dan dianalisa bagaimana proses kerja jaringan ini untuk mendukung tersedianya layanan Instant Messaging dengan kualitas layanan yang baik.

Skripsi ini membahas mengenai perancangan suatu jaringan IMS yang mampu mendukung pengadaan layanan Instant Messaging dan kemudian menganalisa apakah jaringan berbasis IMS ini sudah mampu menyediakan sistem telekomunikasi yang baik dan benar sesuai dengan standar internasional *Quality of Service* (QoS) ITU-T.

1.2 PERUMUSAN MASALAH

Jaringan IMS memiliki berbagai alternatif pemakaian dimana konsep yang cukup sederhana dapat menggunakan OpenIMSCore dan menggunakan instant messaging client untuk menguji penggunaan instant messenger. Melalui elemen Core IMS (HSS & CSCFs) dapat diketahui secara umum message flownya dan dengan menggunakan protocol analyzer, dapat diketahui parameter-parameter

Universitas Indonesia

QoS (Quality of Service) yang menentukan bagus atau tidaknya kualitas jaringan telekomunikasi ini.

Berdasarkan gambaran di atas, dapat dirumuskan masalah-masalah yang diharapkan dapat diketahui melalui penelitian ini:

1. Bagaimanakah konsep jaringan IMS yang diperlukan untuk layanan Instant Messaging?
2. Apa dan bagaimana message flow yang terjadi pada saat registrasi Instant Messaging Client, permintaan panggilan, dan saat terjadi percakapan (chatting)?
3. Bagaimana parameter-parameter QoS jaringan ini saat instant messenger dipakai? Apakah layak pakai?

1.3 TUJUAN PENELITIAN

Penelitian analisa kerja jaringan IMS dengan menggunakan aplikasi instant messenger ini bertujuan untuk:

1. Mengetahui elemen-elemen yang terlibat dalam penyusunan jaringan telekomunikasi menggunakan IMS.
2. Memahami message flow sebagai tahap terjadinya proses telekomunikasi memanfaatkan jaringan IMS.
3. Mengetahui kehandalan performa jaringan IMS untuk mendukung penyediaan layanan Instant Messaging.

1.4 BATASAN MASALAH

Pada penelitian ini, dilakukan tahap pembangunan jaringan IMS dan penggunaan Instant Messaging Client secara internal jadi belum mampu menggambarkan keadaan di luar jaringan internal tersebut. Pengukuran performa IMS dilakukan pada saat proses *chatting* terjadi sehingga perhitungan *delay* tidak termasuk pengiriman paket-paket lain di luar pengiriman pesan IM.

1.5 METODE PENELITIAN

Metode penelitian yang digunakan dalam penyusunan skripsi ini meliputi:

1. Pendekatan dari tinjauan pustaka, yaitu dengan melakukan studi literatur dari buku-buku pustaka, *website*, dan modul instalasi dari perangkat yang digunakan
2. Pendekatan diskusi dengan pembimbing skripsi ataupun teman-teman yang berkaitan dengan topik bahasan skripsi.
3. Perancangan perangkat lunak.
4. Pengujian dan pengambilan data.

Penelitian dilakukan melalui prosedur sebagai berikut:

1. Pembangunan Jaringan OpenIMSCore pada komputer server.
2. Penginstalan IMS Client pada beberapa komputer client.
3. Registrasi IMS Client pada IMS Core.
4. Pembangunan skenario jaringan untuk penelitian.
5. Inisiasi percakapan (chatting) menggunakan fasilitas Instant Messenger yang disediakan IMS Client.
6. Pengukuran QoS jaringan IMS menggunakan Wireshark.
7. Analisa hasil pengukuran QoS.

1.6 SISTEMATIKA PENULISAN

Penulisan Skripsi ini dibagi menjadi beberapa bagian:

BAB 1 PENDAHULUAN

Bab 1 berisi mengenai gambaran umum mengenai pembahasan seminar ini. Bagian ini mencakup latar belakang pengambilan tema, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan penelitian.

BAB 2 LANDASAN TEORI

Bab 2 membahas mengenai teori-teori yang mendasari penelitian skripsi ini, seperti penggunaan protocol SIP (*Session Initiation Protocol*), peran jaringan IMS dalam teknologi telekomunikasi, pembahasan elemen IMS Core, pembahasan layanan Instant

Messaging, parameter-parameter QoS, dan metode analisa yang digunakan pada pengukuran performansi jaringan IMS.

BAB 3 INSTALASI IMS & PROSES INSTANT MESSAGING

Bab 3 membahas mengenai cara instalasi IMS Core dan IMS Client kemudian proses registrasi untuk dilanjutkan pada skenario pembangunan jaringan yang akan digunakan untuk penelitian . berikutnya dibahas mengenai inisiasi panggilan Instant Messaging, dan sampai terjadinya proses *chatting*.

BAB 4 PENGUKURAN DAN ANALISA PARAMETER QoS

Bab 4 berisi mengenai pengukuran dan analisa hasil pengukuran parameter QoS menggunakan Wireshark.

BAB 5 KESIMPULAN

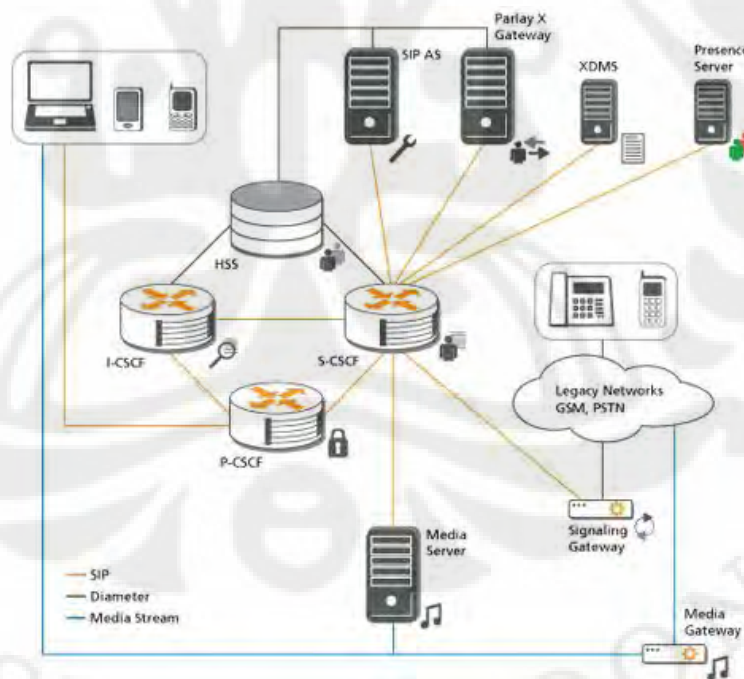
Bab 5 berisi mengenai kesimpulan dari analisis parameter QoS ini untuk menentukan layak tidaknya sistem jaringan IMS digunakan untuk menyokong fasilitas Instant Messenger.

BAB 2

LANDASAN TEORI

Sejak kemunculan spesifikasi IMS pada 3GPP Release 5, IMS didesain untuk mengevolusi jaringan UMTS yang kemudian menjadi komponen inti dari 3G, TV kabel, dan jaringan Next Generation Fixed Telecom. Pada prosesnya, sebagai bagian dari evolusi jaringan *core* dari circuit-switching menjadi packet-switching, IMS yang merupakan sistem berbasis IP didesain untuk membantu operator seluler untuk menyediakan layanan yang interoperabilitasnya tinggi, dengan biaya yang efektif di atas arsitektur jaringan yang memanfaatkan fleksibilitas internet.[1]

Rencana utama dari arsitektur jaringan telekomunikasi ini adalah menempatkan IMS pada *session layer* dari OSI Layer. Dari layer ini, IMS bertugas untuk melakukan *call control & signaling*.



Gambar 2.1 Open IMS Core pada Open IMS Playground

Sumber:http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osi_ms/index.html

2.1 Session Initiation Protocol (SIP)

Protokol SIP merupakan protokol untuk melakukan sambungan, memelihara, dan memutuskan *multimedia session* melalui internet. SIP dikembangkan oleh Internet Engineering Task Force (IETF) sebagai protokol yang tidak mendefinisikan sesi apa yang hendak dilakukan, melainkan hanya melakukan inisiasi saja. SIP merupakan *text-encoded-based protocol* seperti yang digunakan dalam protokol Hyper Text Transport Protocol (HTTP) pada *web-browsing* dan Simple Mail Transport Protocol (SMTP) pada *email*. [2]

SIP dapat memenuhi empat kebutuhan dasar pada komunikasi, yaitu:

1. User Location Services

Fungsi untuk mengetahui identitas dan letak user.

2. Session Establishment

Fungsi untuk menginisiasikan suatu sesi panggilan.

3. Session Participant Management

Fungsi untuk mengontrol sesi yang hendak masuk maupun saat hendak memutus panggilan.

4. Limited Feature Establishment

Fungsi ini digunakan untuk mendeteksi kapabilitas dari devais yang digunakan sehingga ketika suatu devais tidak memungkinkan untuk melakukan panggilan, maka SIP akan memberikan pemberitahuan.

2.1.1 Komponen-komponen SIP :

Pada dasarnya, SIP dapat digunakan pada komponen-komponen berikut ini:

1. User Agent

User agent merupakan end-user device, seperti PC, SIP phone, dll. User Agent dapat dibedakan menjadi dua jenis yaitu User Agent Client dan User Agent Server. User Agent Client merupakan User

Agent yang memulai message (request) User Agent Server merupakan User Agent yang menerima request tersebut dan membalasnya (respond).

2. Registrar Server

Registrar Server merupakan database yang menyimpan semua lokasi dari User Agent yang berada pada domainnya.

3. Proxy Server

Proxy Server menerima request dari User Agent, kemudian melakukan query ke Registrar Server untuk mendapatkan informasi alamat dari tujuan, kemudian meneruskan request tersebut ke User Agent tujuan.

4. Redirect Server

Redirect Server meneruskan request ke eksternal domain. Redirect Server dan Registrar Server dapat berada di hardware yang sama dengan Proxy Server.

2.1.2 SIP Request/Method dan Response

SIP menggunakan metode *request* dan *response* dalam membangun call session. *Request* dan *response* merupakan komunikasi antara *user agent*, proxy dengan *user agent*, atau *user agent* dengan proxy untuk membuat suatu session.

2.1.2.1 Request/Method

Ada enam *request* dasar yaitu:

1. INVITE

INVITE berfungsi untuk melakukan *session setup*.

2. ACK

ACK merupakan jawaban final terhadap INVITE.

3. BYE

BYE berfungsi untuk menghentikan session yang sedang terjadi antar *user agent*.

4. CANCEL

CANCEL berfungsi untuk membatalkan perintah INVITE yang belum mendapatkan respons dari *user agent* tujuan.

5. REGISTER

REGISTER berfungsi untuk melakukan registrasi URI user ke proxy server yang menjadi domainnya.

6. OPTIONS

OPTIONS berfungsi untuk member informasi mengenai kapabilitas dari devais SIP pemanggil maupun penerima.

2.1.2.2 Response

Request message akan mendapat tanggapan berupa *response message*, respons juga memiliki 6 macam, yaitu:

1. 1XX merupakan *informational reponse*
2. 2XX merupakan *success reponse*
3. 3XX merupakan *redirect reponse*
4. 4XX merupakan *request reponse*
5. 5XX merupakan *server reponse*
6. 6XX merupakan *global reponse*

2.1.3 Session Description Protocol (SDP)

Dalam melakukan *session establishment* SIP tidak mendeskripsikan sesi multimedia yang akan dibangun, fungsi tersebut dijalankan oleh protokol tersendiri yang disebut dengan *session description protocol* (SDP). SDP adalah protokol layer aplikasi yang berfungsi untuk memberikan deskripsi tentang suatu sesi multimedia[3], deskripsi tersebut antara lain mengenai :

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

Secara umum, protokol SDP digunakan pada saat melakukan *session announcement* serta *session invitation*. Informasi yang diberikan oleh sebuah pesan SDP antara lain adalah nama *session* dan tujuan penggunaan *session*, waktu

aktif dari sebuah *session*, jenis media yang digunakan, format media, dan informasi untuk menerima media tersebut. Berikut adalah isi dari pesan SDP, *field-field* yang bersifat *optional* diberi tanda *[4].

Session Description

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

One or more time description

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

Time description

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

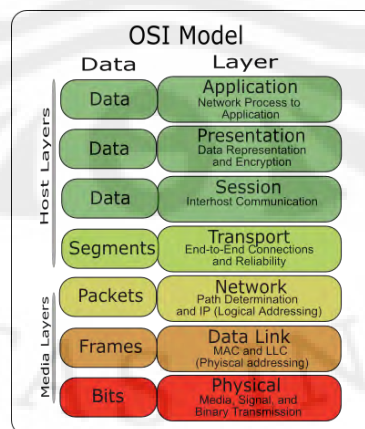
Media description

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

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

2.1.4 SIP Pada Instant Messaging

Untuk mempermudah penjelasan mengenai peran SIP pada instant messaging, perlu kita lihat bagan OSI layer berikut.



Gambar 2.2 OSI Layer

Sumber: http://www.webstepbook.com/supplements/slides/images/osi_model.png

SIP merupakan protocol yang terletak pada *Session Layer* sehingga peran SIP untuk menginisiasi suatu perintah sangat besar pada jaringan IMS. Namun, peran SIP lebih besar dari sekedar melakukan inisiasi sesi percakapan.

Instant Messaging pada dasarnya merupakan pertukaran informasi yang terjadi seperti halnya SMS (Short Message Service) pada teknologi selular sehingga dimaksudkan bahwa informasi yang dikirimkan pada Instant Messaging memang tidak terlalu panjang seperti *e-mail*. Pada IM juga sering dijumpai menu *presence* dimana fungsi *presence* adalah untuk menunjukkan keberadaan rekan (*buddy*) yang juga sedang menggunakan IMS Client.

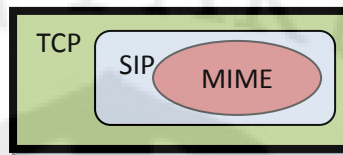
Pada protocol SIP terdapat perpanjangan yang memungkinkan aplikasi Instant Messaging menggunakan SIP sebagai media pengirimannya [5]. Sebuah SIP request dengan metode MESSAGE akan terlihat seperti berikut ini:

```
MESSAGE sip: marks@open-ims.test
Via: SIP/2.0/TCP mkt12.flexanet.com;branch=z8hG4bK123dsghds
Max-Forwards: 69
To: Marks <sip:marks@open-ims.test>
From: Alice <sip:alice@open-ims.test>;tag:49583
Call-ID: asd88asd77a@mkt.12.flexanet.com
CSeq: 1 MESSAGE
Content Type: text/plain
Content Length: 30
Hi Mark, this is an IM example
```

Message Header dari MESSAGE request tidak jauh berbeda dengan SIP request lainnya. Pada bagian *Message body* bisa berisi apapun yang berbasis MIME (Multipurpose Internet Mail Extensions). Guna MIME sendiri adalah untuk memperpanjang format teks maupun non teks. Singkatnya, MIME mampu menambahkan kapasitas SIP yang biasanya hanya digunakan untuk inisiasi percakapan untuk digunakan sebagai media transport data. Pada OSI layer, MIME terletak pada *Presentation Layer*.

Transaksi yang menggunakan SIP MESSAGE ini mengikuti jalur yang sama dengan SIP yang digunakan sebagai *signaling*. Hal ini menjadi penting

karena memungkinkan terjadinya konflik antara keduanya sehingga terjadi kongesti pada jalur SIP. Untuk mencegah hal tersebut, ukuran informasi yang dapat dikirimkan oleh SIP memang sangat dibatasi dan protokol transport seperti TCP dengan mekanisme kontrol terhadap kongesti diperlukan.

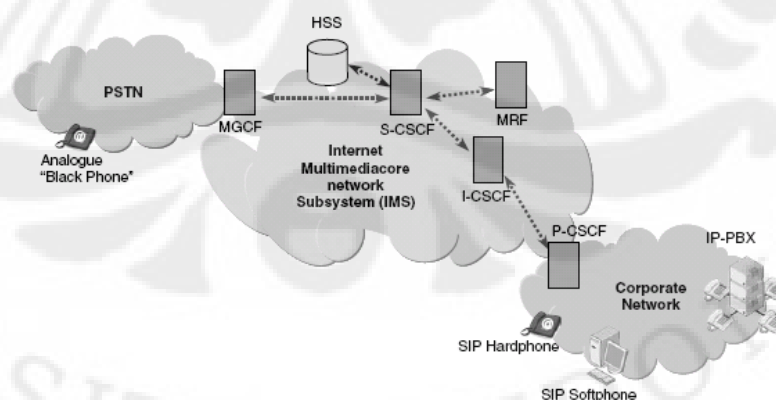


Gambar 2.3 Bagan Paket SIP Message

2.2 IP Multimedia Subsystem (IMS)

IMS (IP Multimedia Subsystem) merupakan sistem yang mampu menggantikan jaringan inti (Core Network) yang diperlukan untuk menyokong pelayanan multimedia [6]. IMS memiliki kedua elemen utama yaitu untuk *signaling* dan media transport. Penggunaan IMS juga memberikan keuntungan berupa kemudahan manajemen karena administrator jaringan cukup menangani satu jaringan IP, bukan jaringan-jaringan IP yang terpisah untuk setiap layanan.

IMS user melakukan koneksi dengan jaringan IMS menggunakan Session Initiation Protocol (SIP) yang berbasis Internet Protocol (IP) sedangkan untuk media yang lebih lama seperti jaringan tetap Public Switched Telephone Network (PSTN) atau jaringan nirkabel GSM dan sejenisnya, diperlukan suatu Media Gateway (MGW) untuk melakukan koneksi dengan jaringan IMS.

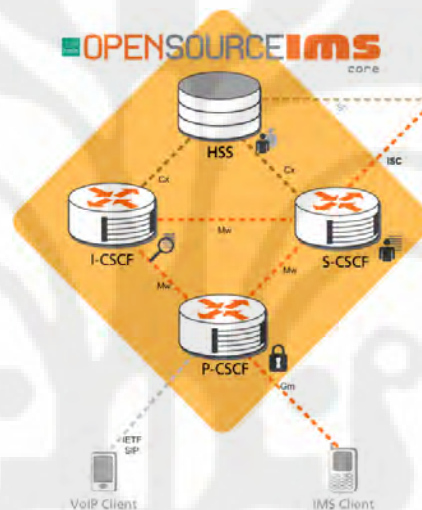


Gambar 2.4 Elemen-elemen IMS

Sumber: Juliet Bates, *et al.*, *Converged Multimedia Network* (John Wiley & Sons, 2006), hal 159

Pada model IMS ini terdapat Call Session Control Function (CSCF) yang melakukan control antara jaringan IMS dengan User Equipment (UE) pada jaringan akses. Di sisi lain, terdapat Media Gateway Control Function (MGCF) yang melakukan control terhadap Gateway menuju terminal lain, dalam hal ini PSTN. Media Resource Function (MRF) berfungsi sebagai jalur informasi dari sumber lain jika dibutuhkan.

IMS Core sendiri merupakan elemen-elemen dasar untuk membangun sistem IMS yang esensial. IMS Core terdiri dari Home Subscriber Server (HSS), Proxy CSCF (P-CSCF), Interrogating CSCF (I-CSCF), dan Serving CSCF (S-CSCF). Elemen-Elemen Core IMS ini memiliki peran yang spesifik.



Gambar 2.5 Core IMS

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

Fungsi dari masing-masing elemen ini akan dipaparkan pada bagian selanjutnya.

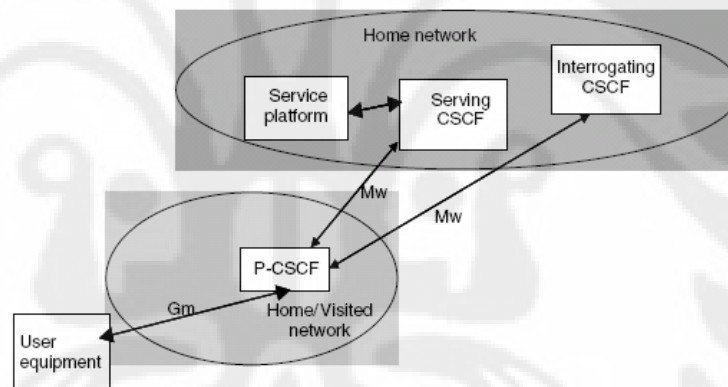
2.2.1 Home Subscriber Server (HSS)

Setiap pelanggan IMS memiliki satu atau lebih identitas pribadi (Private User Identity) yang dimasukkan ke dalam HSS. Identitas ini berlaku secara global dan diketahui di seluruh negara. Identitas ini diberikan pada saat pendaftaran dan memberikan banyak informasi seperti informasi IMS service entitlement, otorisasi, dan akuntansi. S-CSCF harus memiliki dan memasukkan identitas masing-masing pelanggan yang menjadi tanggung jawabnya. Profil pelanggan kemudian diunduh dari HSS ke S-CSCF.

Suatu nomor identitas dapat direpresentasikan kepada satu atau beberapa individu maupun kepada satu atau beberapa layanan sehingga misalkan suatu nomor PSTN dijadikan sebagai nomor identitas untuk beberapa orang yang menggunakan PSTN tersebut.

2.2.2 Proxy Call Session Control Function (P-CSCF)

Suatu P-CSCF merupakan titik masuk untuk pelanggan menuju IMS. P-CSCF adalah aplikasi server yang peran utamanya adalah untuk mengirimkan permohonan SIP kepada P-CSCF atau S-CSCF yang lebih "dekat" dengan tujuan. Jika pelanggan berada dalam keadaan *roaming* (di luar server asalnya) maka perangkat Bergeraknya akan melakukan akses ke IMS melalui P-CSCF terdekat dengan jaringan yang dikunjungi dan jaringan home akan melakukan fungsi IMS lainnya.



Gambar 2.6 CSCFs

Sumber: Juliet Bates, *et al.*, *Converged Multimedia Network* (John Wiley & Sons, 2006), hal 162

2.2.3 Interrogating Call Session Control Function (I-CSCF)

Ketika seorang pengguna akan melakukan registrasi untuk menggunakan layanan IMS, P-CSCF akan meneruskan permintaan SIP registrasi dari pengguna kepada I-CSCF di home domain.

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

CSCF yang sesuai telah terpilih, I-CSCF akan menginformasikan S-CSCF yang harus dipilih oleh pengguna berdasarkan data di HSS.

HSS menyimpan informasi mengenai profil pengguna dan bila terdapat beberapa HSS maka I-CSCF dan S-CSCF akan mencari HSS yang tepat dari Subscription Locator Function (SLF) yang akan memberikan informasi mengenai kapabilitas pengguna untuk mendapatkan layanan tersebut. Bila didapatkan HSS yang sesuai maka HSS tersebut akan menyediakan informasi dari pengguna untuk keperluan akses layanan.

2.2.4 Serving Call Session Control Function (S-CSCF)

Telah dijelaskan sebelumnya bahwa S-CSCF menyediakan proses registrasi layanan kepada pengguna. Seketika setelah I-CSCF memilih S-CSCF yang sesuai dan menyediakan nama S-CSCF tersebut kepada P-CSCF, baik P-CSCF dan S-CSCF yang sesuai akan bertukar informasi untuk proses lebih lanjut. Setelah itu, I-CSCF dapat dihilangkan dari jalur signaling.

Prinsip kerja S-CSCF adalah untuk menyediakan pengguna IMS kepada akses menyeluruh terhadap layanan yang disediakan. Arsitektur IMS adalah berdasarkan prinsip bahwa control layanan untuk HSS (termasuk untuk pelanggan yang di luar jaringan home) berada pada jaringan Home sehingga disinilah S-CSCF selalu diletakkan.

Ketika pelanggan menggunakan nomor telepon dengan format E.164 maka nomor ini akan dikonversi menjadi SIP URI oleh S-CSCF menggunakan mekanisme translasi ENUM DNS. Jika translasi ini gagal, maka pelanggan akan menerima pemberitahuan kegagalan tersebut.

2.3 Layanan Instant Messaging

Keberadaan internet sudah merevolusi cara manusia berkomunikasi, surat menyurat diganti dengan *email* yang lebih cepat dan efisien. Akan tetapi, terkadang *email* masih kurang cepat untuk berkomunikasi karena seseorang mungkin ingin mengetahui keberadaan orang tersebut untuk berbicara secara langsung. Atas dasar ini, muncullah layanan Instant Messaging yang

memungkinkan kita mengetahui keberadaan seseorang lalu mengobrol dengannya.[7]

Layanan Instant Messaging muncul pada awal tahun 1990an dan meledak di pasaran pada November 1996 dimana Mirabilis meluncurkan ICQ yaitu *Instant Messaging Client/Instant Messenger Application* yang berbasis perangkat lunak gratis sehingga dapat digunakan oleh seluruh orang secara cuma-cuma.

Pada saat ini, sangat banyak aplikasi Instant Messenger yang dapat diperoleh secara gratis dengan fitur-fitur yang semakin menarik untuk digunakan. Aplikasi seperti Yahoo! Messenger, MSN Messenger, Xlite, Skype, eBuddy, MyMonster, dan UCT IMS Client dapat menjadi contoh aplikasi yang mendukung layanan Instant Messaging. Fitur-fitur yang disediakan Instant Messenger pun bukan sekedar untuk *chatting* tetapi juga untuk berbagi data, *web* link, melakukan sesi telepon layaknya menggunakan telepon biasa.

Penggunaan Instant Messaging secara umum dapat dilakukan melalui langkah – langkah berikut ini [8]:

1. Jika komputer anda belum memiliki piranti lunak IM, maka anda harus mengunduh piranti lunak IM dulu.
2. Setelah piranti lunak berhasil diunduh, anda harus melakukan instalasi.
3. Setelah piranti lunak IM ada di komputer anda, bukalah piranti tersebut.
4. Begitu anda terhubung dengan server, anda dapat mengetik nama dan password. Jika ini pertama kalinya anda terhubung, anda harus melakukan pendaftaran terlebih dahulu untuk memperoleh *account*, yang dapat digunakan sebagai langkah awal anda menggunakan fasilitas ini. Ketika server melakukan verifikasi terhadap nama dan password anda, maka anda sudah tercatat sebagai member/pengguna fasilitas ini.
5. Setelah itu, komputer anda secara otomatis akan mengirimkan data kepada server berisi informasi koneksi yang ada pada komputer yang anda gunakan (berupa IP address dan number of port).
6. Lalu server akan membuat file sementara yang akan mengkoneksikan informasi untuk anda dan daftar kontak yang anda miliki. Lalu akan dilakukan penyingkronisasian dengan para pengguna yang ada dalam *contact list* anda, dan akan menandai para pengguna yang sedang online.

7. Kemudian anda dapat meng'klik nama orang yang sedang online tersebut, dan sebuah halaman akan terbuka sehingga anda dapat menuliskan pesan. Percakapan dapat dimulai.
8. Ketika percakapan anda telah selesai, anda dapat menutup halaman percakapan anda. Anda dapat mengakhiri koneksi serta keluar dari aplikasi tersebut. Pada saat yang bersamaan, akun anda mengirimkan pesan pada server untuk mengakhiri sesi anda. Server akan mengirimkan pesan kepada setiap rekan anda, bahwa anda telah ada dalam kondisi *logged off*. Sehingga para pengguna IM yang sedang online dan berada di contact list anda, mengetahui anda telah tidak terhubung lagi. Dapat juga disebut *offline*.

Teknologi layanan Instant Messaging pun terus berkembang dan penelitian terakhir menyatakan bahwa layanan ini sedang dikembangkan untuk melakukan fitur "location-aware". Dengan fitur ini, pengguna dapat melacak dan mengetahui dimana keberadaan lawan bicaranya.

Walau bukan termasuk sebagai komunikasi *real-time* dan bukan termasuk cara berkomunikasi paling aman, layanan Instant Messaging memiliki kesederhanaan dalam berkomunikasi yang tidak dimiliki oleh jenis berkomunikasi lainnya seperti bertelepon, surat-menyurat, dan sebagainya. Oleh sebab itu, diperkirakan layanan Instant Messaging akan terus maju dan berkembang sebagai suatu aplikasi yang berpotensi untuk memberikan penghasilan besar bagi pihak-pihak pengembang di dalamnya.

2.4 Parameter QoS Instant Messaging

Parameter QoS (Quality of Service) merupakan parameter-parameter yang menyatakan standar kualitas dari layanan yang diberikan.

Tabel 2.1 Parameter QoS Jaringan

Category	Parameters
Timeliness	Delay Response time Jitter
Bandwidth	Systems-level data rate Application-level data rate Transaction time
Reliability	Mean time to failure (MTTF) Mean time to repair (MTTR) Mean time between failures (MTBF) Percentage of time available Packet loss rate Bit error rate

Sumber: A.H. Muhamad Amin. *VoIP Performance Measurement Using QoS Parameters*.

Malaysia:Tronoh

Dari tabel parameter QoS jaringan di atas terdapat 3 parameter yang biasa digunakan pada penelitian yaitu *delay*, *jitter*, dan *packet loss rate*.

1. *Delay*: ukuran waktu yang diperlukan data sejak dikirimkan dari transmitter sampai diterima pada receiver.
2. *Jitter*: jarak waktu antara paket-paket yang dikirimkan yang berpengaruh pada kemulusan transfer data.
3. *Packet Loss Rate*: ukuran *error rate* yang menggambarkan banyaknya data yang hilang/tidak sampai dalam persentasenya dengan data yang berhasil sampai.

Akan tetapi, pada penelitian Instant Messaging ini, yang dihitung adalah parameter *delay* saja karena pada dasarnya, suatu percakapan instant messaging tidak terlalu membutuhkan ruang yang banyak pada suatu paket. Selain itu, faktor *packet loss* juga jarang dijumpai jika inisiasi percakapan sudah terjadi. Justru yang sering terjadi adalah *delay* dimana *instant messaging* merupakan non-real time data yang seharusnya sanggup memiliki *delay* dibawah 30s [9].

Standar yang digunakan pada pengukuran parameter QoS instant messaging ini adalah ITU-T V-Series yang merupakan standar pengiriman komunikasi data yang direkomendasikan dan sering digunakan untuk modem [10]. Selain itu terdapat *bandwidth* IDSL (144 kbit/s) dan HSDL (1544 kbit/s) sebagai tambahan variasi *bandwidth* yang digunakan.

BAB 3

INSTALASI DAN PROSES INSTANT MESSAGING

Pembangunan suatu sistem jaringan IMS memerlukan elemen IMS Core yang terdiri dari HSS server dan CSCFs server sebagai fondasi jaringannya. Setelah itu, untuk melakukan panggilan dari satu client ke client yang lain diperlukan IMS client di masing-masing perangkat yang sudah teregistrasi ke IMS Core. Proses percakapan *chatting* baru dapat dilakukan setelah masing-masing client melakukan registrasi dan inisiasi panggilan. Ketika kedua pihak sudah berada dalam keadaan *online*, maka panggilan dapat dilakukan dari inisiasi panggilan salah satu pihak ke pihak lainnya.

Berdasarkan skenario singkat di atas, maka dapat diperjelas prosedur yang perlu dilakukan untuk melakukan *chatting* menggunakan jaringan IMS adalah sebagai berikut:

1. Pembangunan Jaringan OpenIMSCore pada komputer server.
2. Penginstalan IMS Client pada beberapa komputer client.
3. Registrasi IMS Client pada IMS Core.
4. Pembangunan skenario jaringan penelitian.
5. Inisiasi percakapan (*chatting*) menggunakan fasilitas Instant Messenger yang disediakan IMS Client.

3.1 Open IMS Core

Open IMS Core merupakan implementasi *open source* dari penggunaan IMS CSCFs dan HSS yang secara bersamaan membentuk elemen inti dari suatu arsitektur IMS/NGN [11]. Penggunaan Open IMS Core ini sangat luas jika dikaitkan dengan testbed arsitektur IMS dan karena sifatnya yang *open source*, Open IMS Core menjadi media yang sering dipakai untuk penelitian dan pengembangan non-komersil.



Gambar 3.1 Open IMS Core

Sumber: http://www.openimscore.org/docs/images/import/logoOpenSource_gross.png

3.1.1 Instalasi Open IMS Core

Langkah pertama dalam pembangunan jaringan IMS Core adalah dengan melakukan instalasi Open IMS Core di dalam komputer server. Berikut adalah panduan untuk melakukan instalasi Open IMS Core pada suatu komputer berbasis Ubuntu (teruji pada Ubuntu 8.04 dan 9.10):

1. Download OpenIMSCore pada PC/Laptop
 - a. Unduh subversion terbaru dengan membuka Menu System > Administration > Software Sources
 - b. Pada tab Ubuntu Software, klik list “Download from” lalu pilih server terdekat (dalam hal ini <http://kebo.vlsm.org/ubuntu>)
 - c. Setelah selesai, klik “Close” dan permintaan pengunduhan subversion terbaru akan muncul, iyaikan.
 - d. Lakukan instalasi subversion package:
 - 1) Masuk pada Menu Application > Accessories > Terminal
 - 2) Ketik `sudo apt-get install subversion`
 - e. Buat folder berikut:


```
sudo mkdir /opt/OpenIMSCore/
```
 - f. Ubah status kepemilikan atas folder ini (ganti username dengan user name anda):


```
sudo chown -R username /opt/OpenIMSCore/
```
 - g. Buat direktori untuk CSCFs dan FHoSS:


```
cd /opt/OpenIMSCore
mkdir ser_ims
mkdir FHoSS
```
 - h. Cek versi terakhir IMS Core dari server Berlios subversion:

svn checkout

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

svn checkout

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

tips: untuk melakukan ini anda membutuhkan koneksi internet. Jika gagal, cek proxy host dan port dengan mengetikan perintah *sudo gedit /etc/subversion/servers*.

2. Instalasi Package yang diperlukan

Package yang diperlukan: sun-java-jdk, mysql-server, libmysqlclient15-dev, libxml2-dev, bind, ant, flex, dan bison. Package ini dapat diinstal melalui Synaptic Package Manager pada menu System > Administration atau melalui perintah di Terminal:

```
sudo apt-get install sun-java6-jdk
```

```
sudo apt-get install mysql-server
```

```
sudo apt-get install libmysqlclient15-dev
```

```
sudo apt-get install libxml2
```

```
sudo apt-get install libxml2-dev
```

```
sudo apt-get install bind9
```

```
sudo apt-get install ant
```

```
sudo apt-get install flex
```

```
sudo apt-get install bison
```

3. Compile

a. Konfigurasi ser_ims:

masuk ke folder `opt/OpenIMScore/ser_ims` lalu masukan perintah

```
sudo make install-libs all
```

Jika ada yang tidak beres, mungkin prasyarat belum terpenuhi

b. Konfigurasi FHoSS:

masuk ke folder `opt/OpenIMScore/FHoSS` lalu masukan perintah

```
sudo ant compile deploy
```


1. Setup Database

Masukan perintah berikut pada Terminal:

```
# 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
```

3.1.2 Konfigurasi DHCP dan DNS

Konfigurasi DHCP dan DNS dapat dilakukan berdasarkan IP server yang hendak digunakan dan juga bergantung kepada keberadaan DNS, apakah di mesin yang sama atau berbeda.

1. Pengaturan pada *localhost* (menggunakan server local)

- a. Jika DNS dijalankan pada mesin yang sama maka edit file `/etc/resolv.conf` dengan perintah:

```
sudo gedit /etc/resolv.conf
```

Lalu tambahkan line berikut:

```
nameserver 127.0.0.1
```

- b. Edit juga file `/etc/hosts` dan tambahkan line:

```
127.0.0.1 localhost
```

- c. File lain yang perlu mengalami edit adalah `/etc/dhcp3/dhclient.conf` dan lakukan uncommment pada baris:

```
prepend domain_name_servers 127.0.0.1;
```

- d. Copy file `open-ims DNS` ke sebuah folder:

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

- e. Tambahkan line berikut ke `/etc/bind/named.conf`

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

- f. Untuk melihat efek dari command tersebut, perlu dilakukan restart.

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

g. Cek apakah semua sudah bekerja dengan mencoba *ping*

```
# ping pcscf.open-ims.test
```

2. Pengaturan menggunakan Proxy tertentu

Jika ingin mengakses Open IMS Core menggunakan perangkat lain, IP *localhost* 127.0.0.1 perlu diganti dengan suatu IP spesifik dimana core IMS tersebut digunakan, misalnya pada penelitian ini digunakan IP 192.168.100.5. Langkah-langkah yang perlu dilakukan untuk mengganti IP ini adalah:

- a. Edit */etc/resolv.conf*
- b. Edit */etc/hosts*
- c. Edit */etc/bind/open-ims.test*
- d. Restart bind: *sudo /etc/init.d/bind9 restart*
- e. Lakukan pengecekan menggunakan perintah: *dig open-ims.test*
- f. Lakukan konfigurasi 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
```

3.1.3 Pengaktifan Perangkat Open IMS Core

Untuk menjalankan Open IMS Core, harus dilakukan pada folder OpenIMSCore untuk itu salinlah file konfigurasi dari OpenIMSCore/ser_ims

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

Jalankan CSCF, masing – masing di terminal yang berbeda (3 terminal):

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

Jalankan FHoSS di terminal yang berbeda namun sebelumnya sesuaikan variable *JAVA_HOME* :

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

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

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

Pastikan bahwa java-6-sun berada pada direktori tersebut, jika tidak maka sesuaikan dengan versi yang ada pada direktori tersebut.

3.1.4 Testing Open IMS Core

Proses terakhir dalam pembangunan jaringan Open IMS Core adalah melakukan pengujian dengan mengakses halaman HSS pada komputer server dengan menggunakan *web browser* seperti Internet Explorer, Mozilla Firefox, atau Opera.

FHoSS seharusnya sudah dapat diakses pada `http://localhost:8080` di web browser. Seharusnya muncul login form seperti berikut ini:



Gambar 3.2 Halaman Otentikasi FhoSS

Setelah itu login-lah dengan akun di bawah ini:

User Name : hssAdmin

Password : hss

Pada jaringan dengan proxy tertentu *localhost* harus diganti dengan proxy server HSS. Jika sudah berhasil dilakukan semua maka jaringan Open IMS Core sudah berhasil dibangun.

3.2 IMS Client

IMS Client dibutuhkan sebagai antarmuka untuk pengguna jaringan IMS yang dalam penerapannya serupa dengan penggunaan Instant Messenger lainnya seperti Yahoo!Messenger atau WindowsLiveMessenger. Perbedaannya adalah IMS Client mendukung penggunaan protocol SIP sehingga IMS Client dapat mengakses jaringan IMS. Beberapa contoh IMS client yang dapat digunakan adalah MyMonster IMS Client yang dikembangkan Fraunhofer Institute FOKUS atau UCT IMS Client yang dikembangkan oleh Communication Research Group di University of Cape Town.

3.2.1 MyMonster



Gambar 3.3 Logo MyMonster

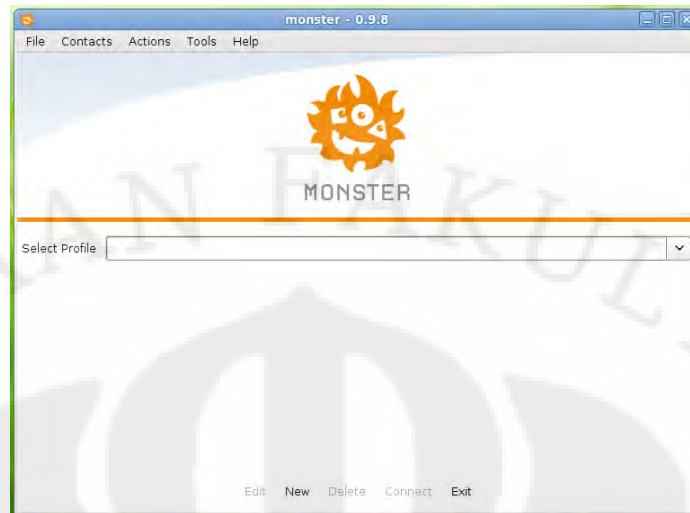
Sumber: http://www.mymonster.org/images/mymonster_logo.png

MyMonster dapat diunduh melalui halaman web <http://www.monster-the-client.org/monster-downloads>. IMS Client ini tersedia untuk 3 macam sistem operasi yaitu Windows, Linux, dan WinMobile. Pembahasan instalasi berikut hanya untuk instalasi di sistem operasi Linux.

1. Unduh MyMonster untuk Linux dari halaman web <http://www.monster-the-client.org/monster-downloads>.
2. Ekstrak file tersebut dengan perintah:


```
Tar -xzf MONSTER-linux32.tgz
```
3. Kemudian, di dalam folder MONSTER-linux32, jalankan MyMonster dengan perintah berikut:


```
java -cp monster-desktop.jar -Dmodule.dirs=.  
de.fhg.fokus.monster.Starter
```
4. MyMonster siap dipakai.



Gambar 3.4 Tampilan Awal MyMonster

3.2.2 UCT IMS Client



Gambar 3.5 UCT Badge

Sumber: <http://uctimsclient.berlios.de/>

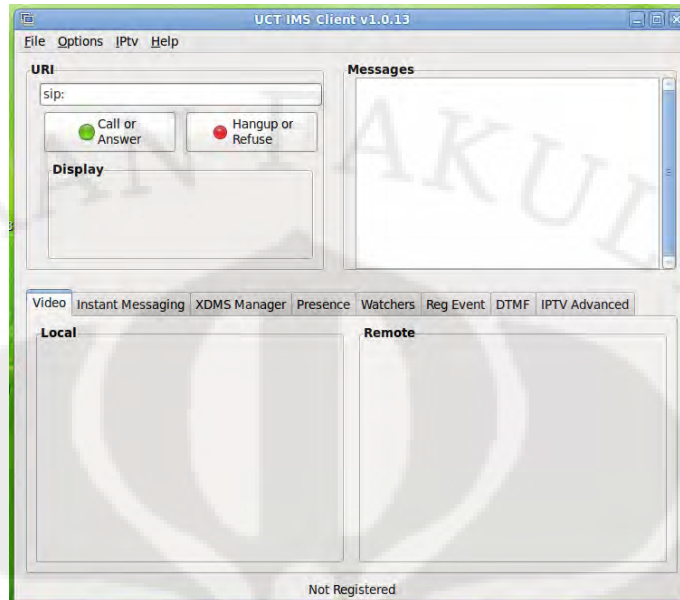
UCT IMS Client merupakan program IMS Client yang dikembangkan di University of Cape Town. Selain menyediakan IMS Client, UCT juga mengembangkan beberapa proyek lain seperti UCT Advanced IPTv, UCT IPTv Streaming Server, UCT Policy Control Framework, UCT Back-toBack User Agent, dan UCT IMS Charging Framework.

Berikut adalah cara melakukan instalasi UCT IMS Client pada sistem operasi Linux:

1. UCT IMS Client dapat diunduh melalui halaman web https://developer.berlios.de/project/showfiles.php?group_id=7844.
2. *Double click* pada file [uctimsclient1.0.13.deb](#)
3. Penuhi prasyarat-prasyarat yang dibutuhkan jika diminta.
4. Jika sudah terinstall maka UCT IMS Client dapat dijalankan dengan perintah:

sudo uctimsclient

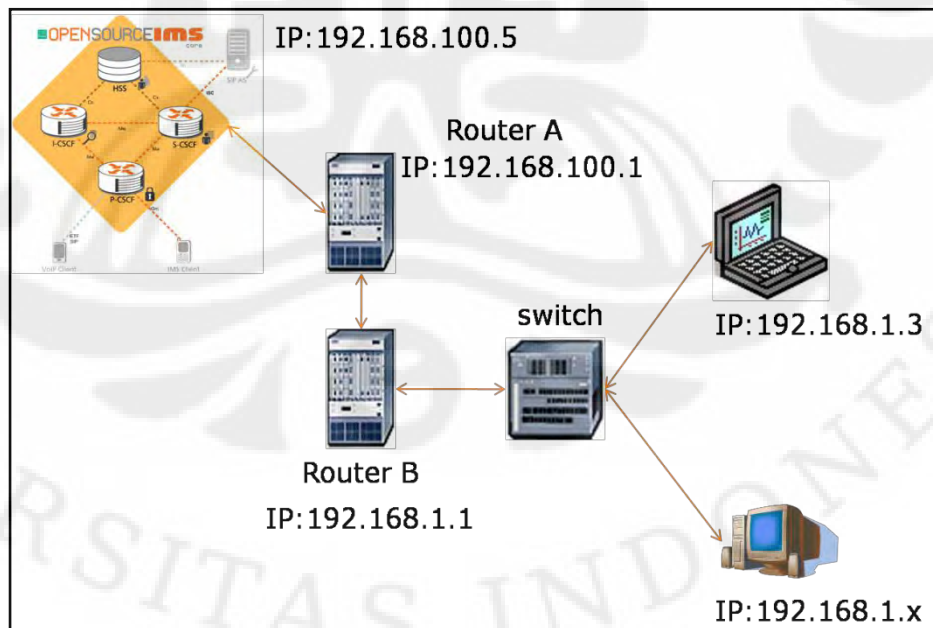
5. UCT IMS Client siap dipakai.



Gambar 3.6 Tampilan Awal UCT IMS Client

3.3 Skenario Jaringan Penelitian

Berikut ini merupakan gambaran jaringan yang digunakan untuk melakukan penelitian.



Gambar 3.7 Skenario Jaringan Penelitian

Berdasarkan skenario di atas, penulis memerlukan sebuah IMS Core server, dua buah *router*, sebuah *switch*, dan dua buah komputer/laptop yang digunakan sebagai *client*.

Server IMS Core dibangun dengan menggunakan sebuah laptop sedangkan kedua buah router merupakan router komputer yang menggunakan program Vyatta di dalamnya. Switch yang digunakan adalah switch LinkSys dan kedua *client* yang digunakan sudah memiliki program IMS Client dan Wireshark sebagai *packet analyzer* di dalamnya.

Skenario yang hendak dilakukan adalah dengan membangun jaringan seperti gambar di atas kemudian mengatur *bandwidth* dari kedua buah router dengan sistem *traffic-shaper*. Penggunaan dua router dimaksudkan untuk menggambarkan suatu kondisi dimana IMS Core berada pada jaringan yang berbeda dengan pelanggan-pelanggan yang menggunakan jaringan IMS. Selain itu penggunaan dua router juga dapat memperkecil secara spesifik hubungan (link) antara kedua jaringan.

Setelah jaringan dibangun, maka langkah yang diperlukan selanjutnya adalah mengaktifkan program IMS Client dan Wireshark yang berada di masing-masing *client*. IMS Client yang digunakan pada percobaan ini dikhususkan pada UCT IMS Client untuk menyederhanakan protokol dan komabilitas masing-masing *client*. Berikutnya, dapat dilakukan prosedur registrasi pelanggan dari masing-masing *client* dan inisiasi percakapan sudah dapat dilakukan.

Proses percakapan ini dijalankan serentak dengan berjalannya program *packet analyzer* Wireshark sehingga paket-paket data yang digunakan selama proses *chatting* akan terekam dan menyimpan data-data otentik dari proses percakapan ini. Hasil rekaman Wireshark ini kemudian dapat digunakan untuk menghitung *delay* dan mampu dianalisa kinerjanya.

3.4 Proses Registrasi

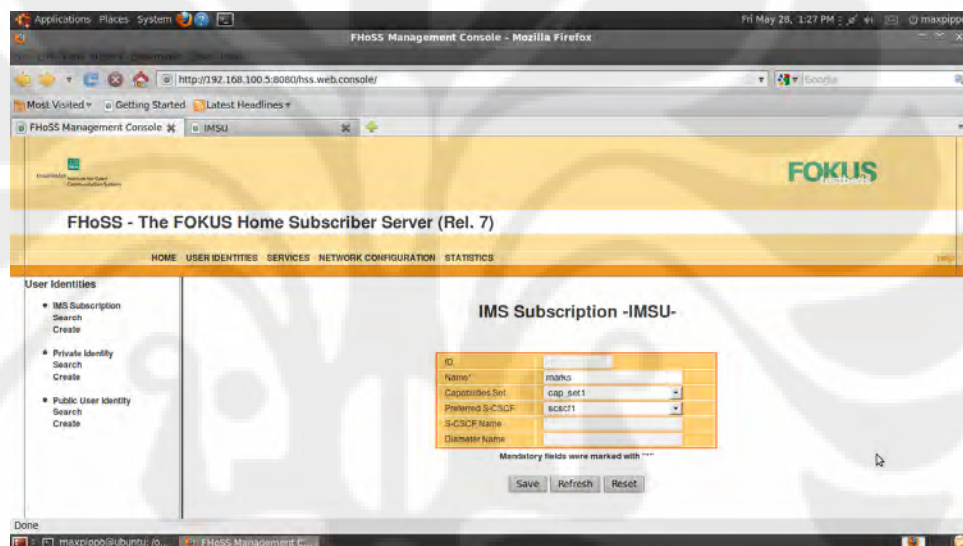
Proses registrasi pelanggan baru harus dilakukan melalui dua tahap utama, yaitu pendaftaran melalui HSS kemudian registrasi menggunakan *client*.

3.4.1 Pendaftaran pelanggan melalui HSS

Pelanggan yang hendak menggunakan jaringan IMS harus terdaftar dalam HSS oleh sebab itu dibutuhkan prosedur untuk mendaftarkan pelanggan yang dilakukan satu kali saja pada awal proses registrasi. Proses pendaftaran ini dilakukan secara manual melalui web browser FHoSS port 8080 pada bagian *user identities*.

1. Registrasi IMSU

Registrasi IMSU merupakan registrasi pertama yang melakukan pendaftaran nama, kapabilitas, SCSCF utama. Setelah dimasukan data yang diperlukan harus disimpan untuk melanjutkan ke langkah selanjutnya.

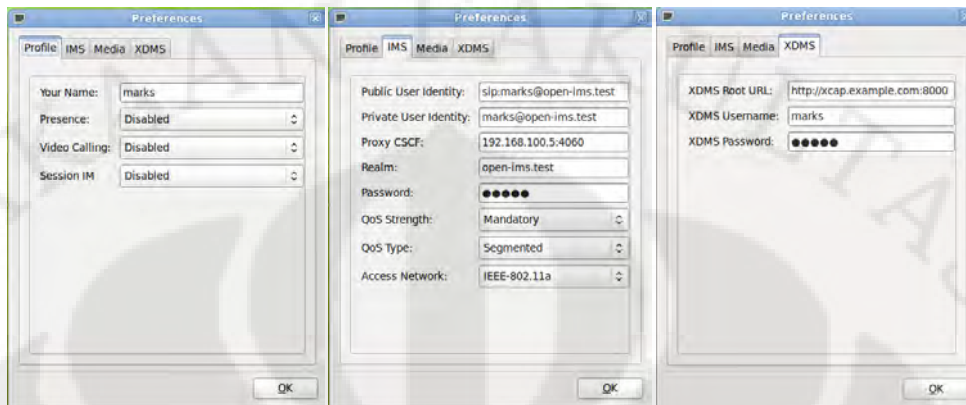


Gambar 3.8 Registrasi IMSU

2. Registrasi IMPI

Registrasi IMPI mendaftarkan identitas privasi user berupa identitas, kode rahasia (password), dan skema otentifikasi. Hal yang perlu diperhatikan adalah penambahan @open-ims.test pada nama di bagian identitas, pemberian *secret key*, dan default Digest-MD5 sedangkan untuk *alternative schemes* dipilih pada kotak *all*.

Setelah membuka aplikasi UCT IMS Client ataupun MyMonster, kita perlu melakukan pengaturan pada bagian *preferences/edit*.



Gambar 3.11 *Preferences* pada UCT IMS Client

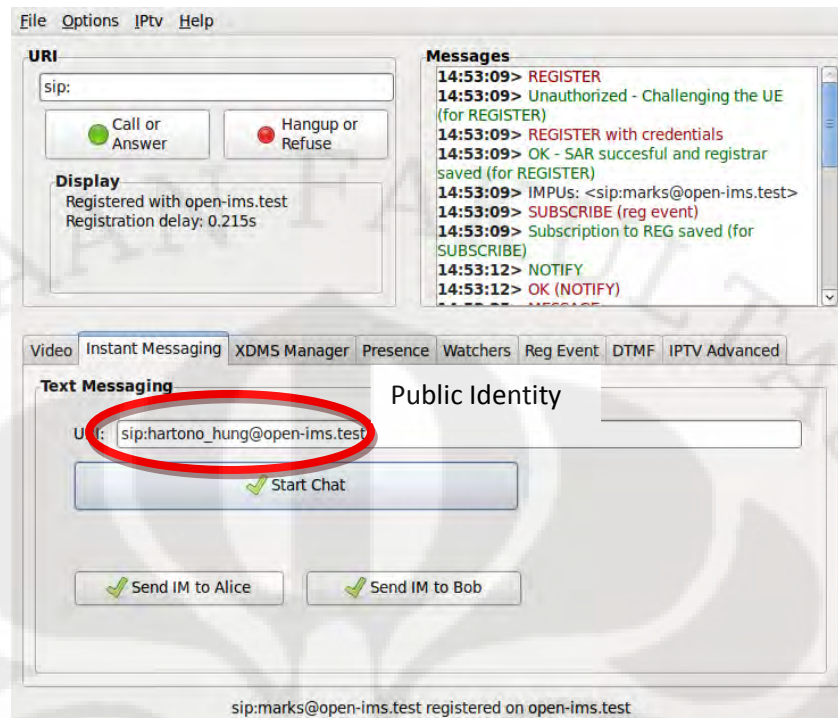
Walau tampilan pengaturan masing-masing IMS Client berbeda-beda, tapi pada dasarnya hal yang perlu diperhatikan adalah:

1. Display Name
2. Public User Identity dan Private User Identity
3. Domain
4. Secret Key/Password
5. PCSCF dan portnya
6. XDMS User Name & Password

Setelah itu, cukup lakukan registrasi yang ditunjukkan dengan pilihan Register atau Connect.

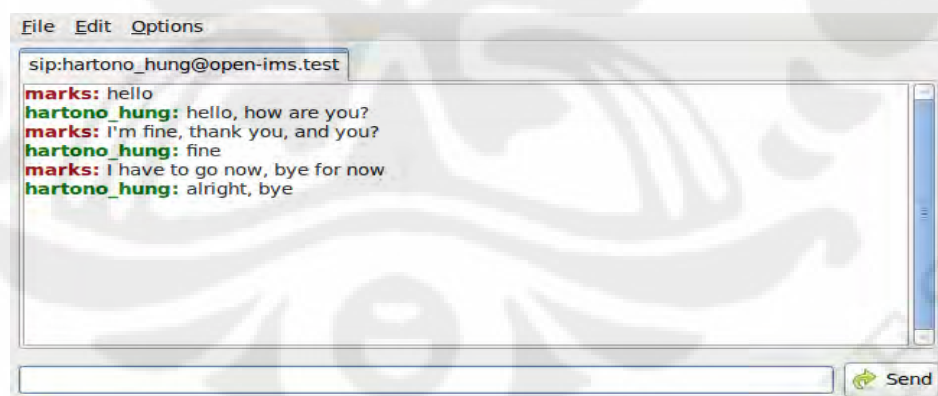
3.5 Inisiasi Percakapan (Chatting)

Percakapan dapat dilakukan bila kita sudah teregistrasi pada HSS, sudah memiliki IMS Client dan menjalankan Client ini pada devais ini. Hal berikutnya yang penting adalah mengetahui alamat lawan bicara yang hendak kita ajak untuk bercakap-cakap. Pada IMS Client, identitas yang perlu diketahui untuk melakukan suatu panggilan adalah Public Identity orang tersebut.



Gambar 3.12 Inisiasi Panggilan dengan Public Identity

Setelah mengetahui Public Identity maka panggilan dapat dilakukan dengan memilih menu “Start Chat” maka akan terbuka kotak dialog yang disiapkan untuk percakapan anda dengan lawan bicara anda.



Gambar 3.13 Dialog Box untuk Proses *Chatting*

BAB 4

PENGUKURAN DAN ANALISA PARAMETER QoS

Proses *chatting* yang dilakukan harus direkam menggunakan program Wireshark. Adapun program Wireshark ini mampu melakukan perekaman (*capturing*) terhadap paket-paket yang telah dikirimkan atau diterima pada suatu protokol di jaringan tertentu.

4.1 Prosedur dan Variabel

Prosedur pengambilan data adalah dengan menggunakan dua Wireshark yang terdapat pada dua client yang hendak menjalankan proses *chatting*. Pertama kali, *bandwidth* diatur melalui kedua router yang digunakan menggunakan standar ITU-T V-Series, IDSL, dan HDSL. Peran variabel *bandwidth* disini adalah sebagai variabel kontrol yang bervariasi. Berikut adalah keenam standar *bandwidth* yang digunakan:

1. V.22 (1,2 kbit/s)
2. V.32 (9,6 kbit/s)
3. V.34 (28,8 kbit/s)
4. V.92 (56 kbit/s)
5. IDSL (144 kbit/s)
6. HDSL (1544 kbit/s)

Dari variabel kontrol di atas, akan diperoleh variabel-variabel terikat. Variabel terikat yang menjadi data adalah berupa pewaktuan dan *message flow*. *Message flow* digunakan sebagai gambaran proses penggunaan jaringan pada alur percakapan ini sedangkan pewaktuan akan dimasukkan ke dalam tabel untuk diproses melalui perhitungan.

Selain variabel kontrol dan terikat, terdapat kondisi penelitian yang harus disamakan dalam setiap pengambilan data yaitu lingkungan. Dalam hal ini lingkungan mencakup jenis IMS Client dan percakapan. Oleh sebab itu, percakapan yang digunakan dalam setiap pengambilan data dibuat sama sehingga lebih memudahkan untuk menentukan waktu pengiriman dan penerimaan pesan.

Percakapan yang menjadi bagian skenario pengambilan data ini adalah seperti berikut ini:

A: Hello

B: Hello, how are you?

A: I'm fine, thank you, and you?

B: Fine

A: I have to go now, bye for now

B: Alright, bye!

Masing-masing percakapan di atas akan dikirimkan sebagai paket SIP sehingga untuk masing-masing *bandwidth* dapat diukur 6 data.

Setelah menentukan semua variabel yang dipakai dalam penelitian ini, maka pengambilan data yang akan dianalisa sudah bisa dilakukan.

4.2 Pengambilan & Pengolahan Data

Data diambil sebanyak 3 kali untuk setiap perubahan *bandwidth* (variabel kontrol). Data diambil berdasarkan waktu yang dibutuhkan dari pengiriman paket pada sisi salah satu client sampai menerima *success response* 200 OK. Karena pengiriman informasi berasal dari dua arah, maka dibutuhkan dua data Wireshark dimana pengiriman untuk nomor ganjil (1, 3, dan 5) menggunakan data dari Wireshark client A sedangkan untuk nomor genap, diambil dari Wireshark client B.

Proses perhitungan *delay* adalah sejak data dikirimkan dari salah satu client sampai didapatkan [SIP:status: 200 OK](#) pada client pengirim.

4.2.1 V.22

Pada V.22, proses registrasi tidak dapat dilakukan karena bitrate yang terlampaui rendah (1,2 kbit/s). Ketika proses registrasi dilakukan, selain pesan REGISTER terlalu lambat untuk dikirim, respon yang diberikan pun menjadi terlambat. Oleh sebab itu, dengan standar V.22, panggilan *chatting* tidak dapat dilakukan dan tidak terukur.

4.2.2 V.32

Tabel 4.1 Data I V.32

Data I	Δ (delay)
1	0,284407 s
2	0,681296 s
3	3,409758 s
4	3,499217 s
5	2,477950 s
6	-
Σ/n (Mean)	2,070526 s

Tabel 4.2 Data II V.32

Data II	Δ (delay)
1	1,102773 s
2	0,117939 s
3	2,346759 s
4	10,103615 s
5	2,179578 s
6	5,466320 s
Σ/n (Mean)	3,552831 s

Tabel 4.3 Data III V.32

Data III	Δ (delay)
1	0,891839 s
2	2,408828 s
3	16,914484 s
4	32,616908 s
5	4,716146 s
6	6,846460 s
Σ/n (Mean)	10,732444 s

Sehingga mean *delay* pada *bandwidth* V.32 (9,6 kbit/s) adalah

$$Delay = (2,070526 + 3,552831 + 10,732444) / 3 = 5,451934 \text{ s}$$

4.2.3 V.34

Tabel 4.4 Data I V.34

Data I	Δ (delay)
1	0,073028 s
2	0,177483 s
3	0,144315 s
4	0,212042 s
5	0,115014 s
6	0,166308 s
Σ/n (Mean)	0,148031 s

Tabel 4.5 Data II V.34

Data II	Δ (<i>delay</i>)
1	0,084454 s
2	0,148763 s
3	0,138091 s
4	0,104516 s
5	0,129432 s
6	0,417014 s
Σ/n (Mean)	0,170378 s

Tabel 4.6 Data III V.34

Data III	Δ (<i>delay</i>)
1	0,107967 s
2	0,053355 s
3	0,186215 s
4	0,126010 s
5	1,540224 s
6	1,901967 s
Σ/n (Mean)	0,652623 s

Sehingga mean *delay* pada *bandwidth* V.34 (28,8 kbit/s) adalah

$$Delay = (0,148031 + 0,170378 + 0,652623) / 3 = 0,323677 \text{ s}$$

4.2.4 V.92

Tabel 4.7 Data I V.92

Data I	Δ (delay)
1	0,091353 s
2	0,169288 s
3	0,055587 s
4	0,117352 s
5	0,137742 s
6	0,143207 s
Σ/n (Mean)	0,119088 s

Tabel 4.8 Data II V.92

Data II	Δ (delay)
1	0,014865 s
2	0,062580 s
3	0,137382 s
4	0,055236 s
5	0,257963 s
6	0,015805 s
Σ/n (Mean)	0,090634 s

Tabel 4.9 Data III V.92

Data III	Δ (delay)
1	0,134695 s
2	0,009591 s
3	0,079542 s
4	0,018821 s
5	0,083540 s
6	0,900105 s
Σ/n (Mean)	0,054365 s

Sehingga mean *delay* pada *bandwidth* V.92 (56 kbit/s) adalah

$$Delay = (0,119088+0,090634+0,054365)s/3 = 0,088029 \text{ s}$$

4.2.5 IDSL

Tabel 4.10 Data I IDSL

Data I	Δ (delay)
1	0,014404 s
2	0,205237 s
3	0,083394 s
4	0,072866 s
5	0,089611 s
6	0,136124 s
Σ/n (Mean)	0,100273 s

Tabel 4.11 Data II IDSL

Data II	Δ (<i>delay</i>)
1	0,102994 s
2	0,112129 s
3	0,057876 s
4	0,142853 s
5	0,036935 s
6	0,156003 s
Σ/n (Mean)	0,101465 s

Tabel 4.12 Data III IDSL

Data III	Δ (<i>delay</i>)
1	0,058081 s
2	0,015538 s
3	0,095939 s
4	0,078876 s
5	0,029732 s
6	0,119505 s
Σ/n (Mean)	0,066279 s

Sehingga mean *delay* pada *bandwidth* IDSL (144 kbit/s) adalah

$$Delay = (0,100273+0,101465+0,066279)s/3 = 0,089339 \text{ s}$$

4.2.6 HDSL

Tabel 4.13 Data I HDSL

Data I	Δ (delay)
1	0,079712 s
2	0,071632 s
3	0,081835 s
4	0,064427 s
5	0,041385 s
6	0,061223 s
Σ/n (Mean)	0,066702 s

Tabel 4.14 Data II HDSL

Data II	Δ (delay)
1	0,032295 s
2	0,084731 s
3	0,047058 s
4	0,015404 s
5	0,011178 s
6	0,014863 s
Σ/n (Mean)	0,034255 s

Tabel 4.15 Data III HDSL

Data III	Δ (delay)
1	0,019108 s
2	0,064973 s
3	0,063249 s
4	0,017805 s
5	0,120155 s
6	0,074868 s
Σ/n (Mean)	0,060026 s

Sehingga mean *delay* pada *bandwidth* HDSL (1544 kbit/s) adalah

$$Delay = (0,066702+0,034255+0,060026)s/3 = 0,053661 \text{ s}$$

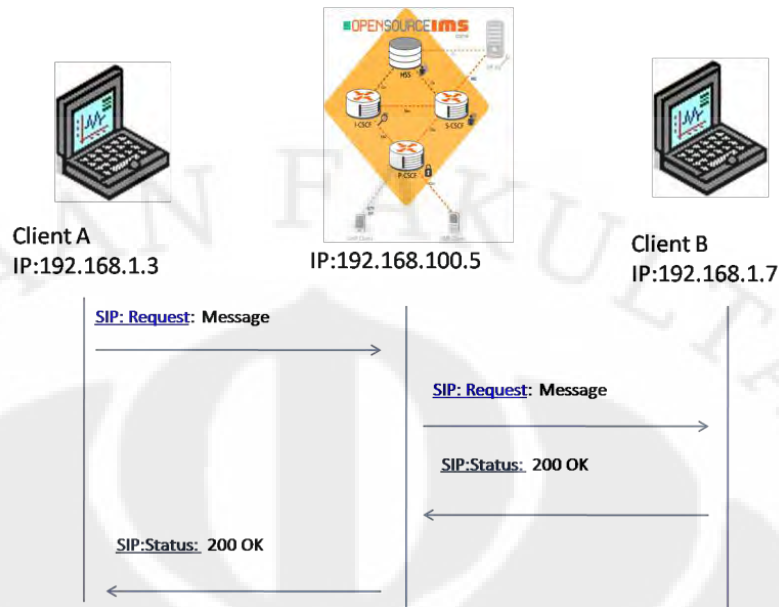
4.2.7 Message Flow

Alur pesan yang terjadi diambil berdasarkan proses registrasi dan percakapan yang berjalan. Pada proses registrasi, message flow yang terjadi adalah sebagai berikut:

Time	192.168.1.3	192.168.100.5	Comment
0.000	Request REGISTER s		SIP: Request: REGISTER sip:open-ims.test
0.146	Status: 401 Unautho		SIP: Status: 401 Unauthorized - Challenging the UE (0 bindings)
0.295	Request REGISTER s		SIP: Request: REGISTER sip:open-ims.test
0.664	Status: 200 OK - SA		SIP: Status: 200 OK - SAR successful and registrar saved (1 bindings)
0.697	Request SUBSCRIBE		SIP: Request: SUBSCRIBE sip:marks@open-ims.test
0.705	Status: 200 Subscri		SIP: Status: 200 Subscription to REG saved
4.682	Request NOTIFY sip		SIP: Request: NOTIFY sip:marks@192.168.1.3:5060
4.683	Status: 200 OK		SIP: Status: 200 OK

Gambar 4.1 Message Flow Registrasi

Sedangkan untuk menyederhanakan penggambaran hasil skematik percakapan (chatting), maka akan digambarkan seperti gambar berikut:

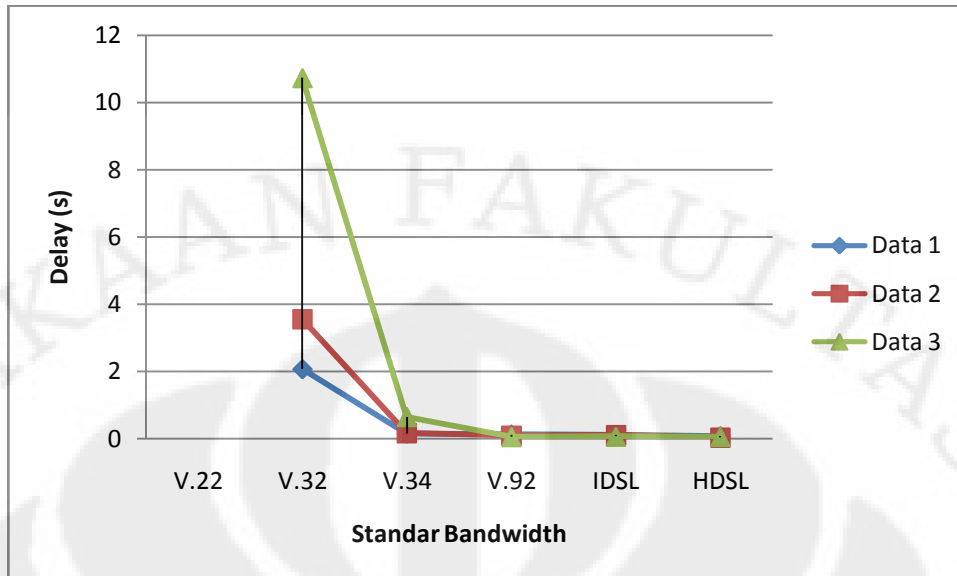


Gambar 4.2 Message Flow Chatting

Pada alur pesan di atas, digambarkan suatu kondisi dimana ada pesan yang dikirimkan oleh client A kepada client B melalui perintah [SIP:request](#): Message. Setelah diterima di client B maka client B akan memberikan respons konfirmasi bahwa pesan sudah diterima dengan [SIP:Status](#): 200 OK. Saat pesan dikirimkan dari client B menuju client A, alur pengiriman pesan seperti di atas juga berlaku hanya arah pengirimannya saja yang terbalik.

4.3 Analisa Delay

Berdasarkan data-data yang diperoleh pada bagian 4.1, dapat dianalisa berbagai pengaruh *bandwidth* yang dibatasi sedemikian rupa sehingga menyebabkan jalur transmisi data terkontrol. Hubungan antara perubahan *bandwidth* dengan *delay* sendiri dapat diilustrasikan sebagai berikut:



Gambar 4.3 Grafik Hubungan *Bandwidth* dengan *Delay*

Pada standar V.22 yang memiliki *bandwidth* 1,2 kbps dapat dikatakan IM belum dapat berfungsi, karena proses registrasi belum dapat dilakukan karena pesan registrasi yang dikirimkan menuju IMS Core sangat lambat, begitu pula respons yang diberikan oleh Server IMS pun tidak sampai ke client yang melakukan registrasi.

Pada standar V.32 yang memiliki *bandwidth* 9,6 kbps, IM sudah dapat dijalankan, proses registrasi pun berhasil dilakukan walaupun memakan waktu yang cukup lama. Dengan rata-rata *delay* 5,451934 sekon, dapat dikatakan bahwa standar V.32 merupakan persyaratan minimum untuk menjalankan fungsi IM pada jaringan IMS Core. Akan tetapi, terdapat sebuah data yang memiliki *delay* lebih dari 30 sekon yang berarti tidak memenuhi standar ITU-T sehingga walau memiliki *delay* rata-rata yang sudah cukup baik, sistem ini belum stabil untuk menjalankan fungsi IM.

Pada standar V.34 yang memiliki *bandwidth* 28,8 kbps, fungsi IM sudah berjalan dengan cukup baik, registrasi dapat dilakukan dengan lancar. Rata-rata *delay* yang didapatkan dari penelitian dengan *bandwidth* ini berkisar pada 323 milisekon sehingga sudah layak pakai untuk digunakan. Kekurangan yang masih ada pada sistem ini terlihat pada saat data yang hendak dikirimkan mengandung informasi yang terlalu panjang. Selain membebani paket, informasi yang terlalu panjang juga cenderung menyebabkan *delay* yang lebih panjang dari biasanya.

Pada standar V.92 yang memiliki *bandwidth* 56 kbps, IM sudah memiliki performa yang baik layaknya IM yang biasa digunakan. Registrasi berjalan secara lancar dan cepat, inisiasi panggilan untuk ajakan *chat* pun berhasil direspon dengan cukup baik. *Delay* rata-rata yang diperoleh sebesar 88 milisekon pun sudah menunjukkan bahwa IM sudah dapat bekerja dengan responsif pada standar ini. Saat menangani informasi yang lebih panjang, walau sedikit menambah *delay* namun tidak terlalu signifikan besarnya sehingga proses *chatting* dapat dinikmati dengan baik.

Pada standar IDSL, *bandwidth* yang digunakan pada hubungan dua router adalah 144 kbps. Pengolahan data dari data-data yang berhasil diukur pada standar ini menunjukkan bahwa *delay* dari *bandwidth* ini memiliki rata-rata berkisar antara 89 milisekon, sedikit lebih besar dari standar V.92. Hal ini menunjukkan anomali karena pada teorinya semakin besar *bandwidth* yang digunakan maka transmisi data akan semakin mudah dan seharusnya *delay* menjadi semakin kecil. Akan tetapi, hal ini bisa saja terjadi mengingat data yang dikirimkan suatu pesan IM memiliki standar tertentu meninjau alur paketnya, memperhitungkan pemrosesan di IMS Server, dan mengasumsikan besar paket adalah sama sehingga waktu *delay* tidak terlalu berpengaruh walau standar yang digunakan semakin baik.

Pada standar HDSL yang memiliki *bandwidth* 1544 kbps, IM berjalan semakin lancar, proses registrasi, inisiasi percakapan, maupun saat terjadi penumpukan data juga sudah dapat ditangani dengan lebih baik. Rata-rata *delay* yang berhasil didapatkan dari pengolahan data adalah 53 milisekon. Dengan demikian, terbukti bahwa semakin besar *bandwidth* yang digunakan maka akan semakin kecil *delay* yang terjadi. Akan tetapi, penurunan nilai *delay* di sini tidak terlalu signifikan dibandingkan standar V.92.

Beberapa faktor yang menyebabkan *delay* pada percobaan ini adalah:

1. Perubahan *bandwidth* yang diatur dari kedua router.
2. Panjang informasi yang mempengaruhi besar paket yang dikirimkan.

Delay yang dianalisa tidak menggambarkan *delay* keseluruhan proses pengiriman paket karena ada *looping presence* yang dapat masuk di antara proses

pengiriman pesan. Dengan kata lain, *delay* yang dihitung dapat mengandung waktu yang diperlukan untuk *looping presence* juga.

Berdasarkan analisa di atas, dapat disimpulkan bahwa standar minimum yang bisa digunakan untuk penggunaan IM adalah standar V.32 yang memiliki *bandwidth* 9,6 kbps. Namun standar ini belum menunjukkan kestabilan terhadap fungsi IM dan jaringan IMS sendiri. Standar optimal yang menunjukkan performansi dan kestabilan yang baik tanpa memboroskan *bandwidth* yang dimiliki adalah standar V.92 yang memberikan *bandwidth* sebesar 56 kbps.

4.4 Analisa Tambahan

Pada penelitian yang dilakukan juga didapatkan beberapa informasi mengenai fitur-fitur Instant Messaging Client yang digunakan.

1. Pengiriman karakter yang dapat berubah menjadi gambar (*emoticon*) dipengaruhi oleh fitur IM Client yang digunakan sehingga pada jaringan IMS tidak dibebankan pengiriman data gambarnya. Fitur *emoticon* berada pada masing-masing IM Client.
2. Pengembangan IM Client untuk melakukan percakapan bersama-sama (*conference*) memungkinkan untuk dilakukan.
3. Maksimum karakter yang dapat dikirimkan oleh satu paket SIP berkisar antara 215 karakter bergantung pada jenis karakternya juga.

BAB 5

KESIMPULAN

Dari keseluruhan pembahasan dalam skripsi ini, dapat disimpulkan beberapa hal yang bisa dikaitkan dengan performansi Layanan Instant Messaging di implementasi Open IMS Core pada jaringan yang digunakan, yaitu:

1. Jaringan Open IMS Core yang terdiri dari HSS, P-CSCF, I-CSCF, dan S-CSCF sudah mendukung untuk layanan Instant Messaging.
2. Dua router yang digunakan pada implementasi ini adalah untuk mengkondisikan bahwa kedua client dan IMS Core berada pada jaringan yang berbeda. Dari kedua router juga diberi perubahan *bandwidth* sebagai variabel control.
3. Standar V.32 dengan *bandwidth* 9,6 kbps dapat menjadi persyaratan minimum berjalannya fungsi IM pada implementasi Open IMS Core karena rata-rata *delay* yang diperoleh sebesar 5,451934 sekon sudah berada di bawah standar yang disarankan ITU-T yaitu lebih kecil dari 30 sekon.
4. *Bandwidth* optimal yang bisa digunakan untuk aplikasi Instant Messenger adalah 56 kbps (V.92). Standar ini dapat dikatakan optimal karena fungsi IM sudah berjalan dengan baik, stabil, dan memiliki *delay* yang sudah cukup rendah.

DAFTAR ACUAN

- [1] *IMS (IP Multimedia Subsystem)*. Diakses 13 April 2010, dari eGPP
<http://www.3gpp.org/article/ims>
- [2] S., R. Kharisma Adhiputra (2009, Desember). *Session Initiation Protocol sebagai Call Control pada Aplikasi VoIP*, Seminar. Depok: Universitas Indonesia.
- [3] Hens, F.J & Caballero, J.M.(2008). *Triple Play* (hal 121). John Wiley & Sons
- [4] M.Handley, V.Jacobson,& C.Perkins.*SDP:Session Initiation Protocol*. July 2006.RFC 4566.
- [5] Hens, F.J & Caballero, J.M.(2008). *Triple Play* (hal 126). John Wiley & Sons
- [6] Juliet Bates, *et al.*(2006). *Converged Multimedia Network* (hal 159-165). John Wiley & Sons
- [7] *How Instant Messaging Works*. Diakses 27 April 2010, dari howstuffworks
<http://communication.howstuffworks.com/instant-messaging.htm>
- [8] *Pesan Instan*. Diakses 27 April 2010, dari Wikipedia
http://id.wikipedia.org/wiki/Pesan_instan
- [9] ITU-T (2001, November). *End-user Multimedia QoS Categories* (ITU-T Rec. G.1010).
- [10] *ITU-T V Series Recommendations*. Diakses 3 Juni 2010 dari Wikipedia.org
http://en.wikipedia.org/wiki/ITU-T_V-Series_Recommendations
- [11] *The Open Source IMS Core Project*. Diakses 11 Mei 2010 dari OpenIMScore.org
<http://www.openimscore.org/>

LAMPIRAN

Lampiran: Paket Pengiriman Pesan

The image shows a Wireshark capture of network traffic. The main pane displays a list of captured packets, with the selected packet (No. 21) expanded to show its details. The details pane shows the following information:

- User Datagram Protocol:** Src Port: sip (5060), Dst Port: dsimeter_iatc (4060)
- Session Initiation Protocol**
 - Request-Line: MESSAGE sip:hartono_hung@open-ims.test SIP/2.0
 - Message Header
 - Via: SIP/2.0/UDP 192.168.1.3:5060;rport;branch=29h64b41139758702
 - Route: <sip:orig@scsf.open-ims.test:6060/>
 - From: "marks" <sip:marks@open-ims.test>;tag=L72474182
 - To: <sip:hartono_hung@open-ims.test>
 - Call-ID: 182015896
 - CSeq: 20 MESSAGE
 - Content-Type: text/plain
 - Max-Forwards: 70
 - User-Agent: UCT IMS Client
 - P-Preferred-Identity: "marks" <sip:marks@open-ims.test>
 - P-Access-Network-Info: IEEE-802.11a
 - Content-Length: 5
 - Message Body
 - Line-based text data: text/plain
 - hello

The packet list pane shows the following details for the selected packet (No. 21):

No.	Time	Source	Destination	Protocol	Info
16	12.307233	192.168.1.3	192.168.100.5	SIP	Request: MESSAGE sip:hartono_hung@open-ims.test
17	12.569972	192.168.100.5	192.168.1.3	UDP	Source port: dsimeter_iatc Destination port: sip
18	12.571386	EdhmaxTe_24;cc:52	Broadcast	ARP	Who has 192.168.1.2? Tell 192.168.1.1
19	13.367928	192.168.1.3	192.168.100.5	SIP	Request: MESSAGE sip:hartono_hung@open-ims.test
20	13.571963	EdhmaxTe_24;cc:52	Broadcast	ARP	Who has 192.168.1.2? Tell 192.168.1.1
21	13.574944	192.168.1.3	192.168.100.5	SIP	Request: MESSAGE sip:hartono_hung@open-ims.test (text/plain)
22	13.594052	192.168.100.5	192.168.1.3	SIP	Status: 200 OK
23	14.571556	EdhmaxTe_24;cc:52	Broadcast	ARP	Who has 192.168.1.2? Tell 192.168.1.1
24	14.616727	192.168.100.5	192.168.1.3	SIP	Request: MESSAGE sip:marks@192.168.1.3:5060;Line=df131df25550b14
25	15.069093	192.168.100.5	192.168.1.3	SIP	Request: MESSAGE sip:marks@192.168.1.3:5060;Line=df131df25550b14
26	15.379718	192.168.1.3	192.168.100.5	SIP	Request: MESSAGE sip:hartono_hung@open-ims.test
27	15.729069	192.168.1.3	192.168.100.5	SIP	Request: MESSAGE sip:hartono_hung@open-ims.test