

Implementasi IMS (IP Multimedia Subsystem) Menggunakan Protokol SIP (Session Initiation Protocol) Pada Jaringan Fakultas Ilmu Terapan

Rafki Altoberi
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
aralvidra02@gmail.com

Giva Andriana Mutiara
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
giva.andriana @tass.telkomuniversity.ac.id

Muh.Fahru Rizal
Fakultas Ilmu Terapan
Telkom University
Bandung, Indonesia
mfrizal @tass.telkomuniversity.ac.id

Abstrak—Awal dari konsep NGN (*Next Generation Network*) adalah Teknologi *softswitch*. Teknologi ini memisahkan *softswitch* dari fungsi aplikasi server yang memungkinkan pengadaan layanan atau aplikasi tanpa mengubah konfigurasi lapisan *transport* maupun lapisan akses di bawahnya. IMS adalah salah satu arsitektur jaringan yang memungkinkan terjadinya pemusatan data dan suara melalui infrastruktur berbasis IP. Penelitian ini bertujuan untuk membangun arsitektur IMS (IP Multimedia Subsystem) dengan menggunakan protokol SIP (*Session Initiation Protocol*) pada jaringan lokal dan mengukur kehandalan implementasi dengan mengukur parameter QoS. Hasil pengujian menunjukkan fitur-fitur seperti *voice call*, *video call*, *sharing file*, dan *messaging* bekerja dengan baik pada perangkat *smartphone* dan *laptop*.

Kata Kunci—IMS, SIP, *video call*, *voice call*, *messaging*

Abstrak— The beginning of NGN (*Next Generation Network*) concept is *softswitch* technology. This technology separates *softswitch* from the function of server application that allows the procurement of services or applications without changing the configuration of transport layer or access layer underneath. IMS is one of network architectures that enable the centralization of data and voice over IP-based infrastructure. This study aims to build IMS (IP Multimedia Subsystem) architecture by using SIP (*Session Initiation Protocol*) protocol on the local network and to measure the reliability of implementation by measuring the QoS parameters. The test results demonstrate that features such as *voice calls*, *video calls*, *file sharing*, and *messaging* work well on *smartphones* and *laptops*.

Keywords—IMS, SIP, *video call*, *voice call*, *messaging*

I. PENDAHULUAN

Awal dari konsep NGN (*Next Generation Network*) adalah teknologi *softswitch*. Teknologi ini memisahkan *softswitch* dari fungsi aplikasi *server* yang memungkinkan pengadaan layanan

atau aplikasi tanpa mengubah konfigurasi *layer transport* maupun layer akses di bawahnya. Teknologi NGN berkembang menuju konvergensi layanan *voice* dan data pada PSTN dan *mobile equipment*. Dengan adanya IMS diharapkan terjadi konvergensi antara layanan *fixed* dan *mobile*.

IMS adalah salah satu arsitektur jaringan yang memungkinkan terjadinya pemusatan data dan suara melalui infrastruktur berbasis IP. Arsitektur IMS merupakan teknologi komunikasi yang dapat menggabungkan antara teknologi *wired* dan *wireless* secara *real time* dan dapat memberikan layanan multimedia secara interaktif. IMS dirancang agar mampu menyediakan layanan multimedia yang lebih kompetitif dengan tingkat mobilitas yang tinggi. IMS juga didesain agar mampu dijalankan tanpa adanya batasan area maupun *domain* yang ada. Pada penelitian ini, akan dilakukan implementasi arsitektur IMS menggunakan protokol SIP di jaringan lokal Fakultas Ilmu Terapan yang bertujuan untuk dapat melakukan komunikasi multimedia interaktif seperti pertukaran data, komunikasi secara *realtime* dan lain sebagainya antar berbagai jenis *client* melalui arsitektur IMS, selain itu juga melakukan pengukuran parameter QoS terhadap kehandalan implementasi.

A. Rumusan Masalah

Adapun rumusan masalah yang dapat diuraikan adalah :

1. Bagaimana mengimplementasikan arsitektur IMS dengan protokol SIP pada jaringan *intranet*?
2. Bagaimana mengimplementasikan client IMS pada *smartphone*, *laptop* dan PC?
3. Bagaimana mengukur kehandalan implementasi sistem?

B. Tujuan

Penelitian ini bertujuan untuk mengimplementasikan arsitektur IMS dengan protokol SIP pada jaringan *intranet* dan mengimplementasikan *client* IMS pada *smartphone* dan *laptop*,

kemudian mengukur kehandalan implementasi dengan mengukur parameter QoS.

C. Batasan Masalah

- Ruang lingkup penulisan dibatasi sebagai berikut:
1. Skenario jaringan dijalankan pada jaringan lokal Fakultas Ilmu Terapan, Universitas Telkom.
 2. Menggunakan *mobile device* dengan OS android dan *laptop* sebagai *client*.
 3. Menggunakan OS ubuntu 10.04 pada *server* IMS.
 4. Menggunakan OpenIMS sebagai *server* IMS.
 5. Hanya menampilkan hasil QoS (*delay, jitter, dan throughput*).

II. LANDASAN TEORI

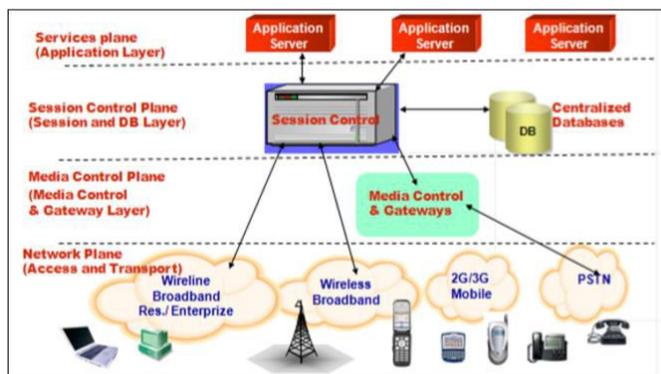
A. IMS (IP Multimedia Subsystem)

IMS (IP Multimedia Subsystem) adalah sebuah arsitektur *framework* untuk mengirimkan layanan internet protokol multimedia. Arsitektur ini dirancang sesuai dengan standar *wireless* oleh 3GPP (3rd Generation Partnership Project), sebagai bagian dari visi untuk berkembang diluar jaringan GSM. Untuk kemudahan dalam mengintegrasikan layanan IMS dengan internet, IMS menggunakan protokol SIP (*Session Initiation Protocol*). Layanan IMS merupakan layanan yang mempunyai jangkauan luas karena keunggulan protokol SIP dalam hal fleksibilitas yang digunakan pada jaringan ini. IMS mendukung *multiple application server* yang dapat memberikan layanan telepon PSTN dan non telepon. IMS mempunyai beberapa kemampuan sebagai lapisan jaringan *packet switched* dan *circuit switched existing*, diantaranya:

1. Dukungan untuk multimedia berbasis IP dengan dukungan *Quality of Service*
2. Pensinyalan berdasarkan protokol SIP
3. Dapat berintegrasi dengan *Public Switch Telephone Network* (PSTN)

IMS meningkatkan kemampuan dari *packet-switched* jaringan bergerak (seperti 3G GSM) dengan mendukung jasa dan aplikasi berbasis IP melalui protokol SIP. Secara efektif, IMS menyediakan suatu arsitektur pemersatu yang mendukung cakupan yang luas dari jasa berbasis IP diatas jaringan *packet* dan *circuit switch*, memanfaatkan perbedaan teknologi akses *wireless* dan *wireline* [1].

Arsitektur tersebut dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur IMS [6]

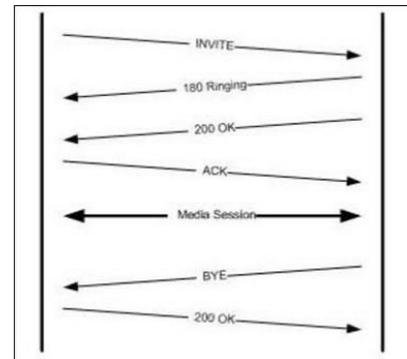
Berikut adalah *layer-layer* yang terdapat pada arsitektur IMS:

1. Server Aplikasi
2. Session Control
3. Transport dan Endpoint.

B. SIP (Session Initiation Protocol)

SIP (*Session Initiation Protocol*) adalah *peer-to-peer signaling* protokol, dikembangkan oleh IETF, yang mengizinkan *end-point*-nya untuk memulai dan mengakhiri sesi komunikasi. Protokol ini didefinisikan pada RFC2543 dan menyertakan protokol lain yang dikembangkan IETF, mencakup *Hypertext Transfer Protokol* (HTTP) yang diuraikan pada RFC 2068, *Simple Mail Transfer Protokol* (SMTP) yang diuraikan pada RFC2821, dan *Session Description Protokol* (SDP) yang diuraikan pada RFC2327. SIP bukanlah protokol yang membawa paket data *voice* atau *video*. Dalam implementasi VoIP, protokol yang dijadikan sebagai media transfer adalah *Real Time Protokol* (RTP). SIP menggunakan protokol UDP pada port 5060, sedangkan RTP menggunakan protokol UDP dengan port yang dinamis.[2].

- SIP Request dan Response. Gambar 2 menunjukkan proses inisiasi sesi pada protokol SIP:



Gambar 2. SIP Request Response

- SIP Request

INVITE yaitu proses mengundang *user agent* yang lain untuk bergabung dalam sesi komunikasi.

ACK yaitu konfirmasi bahwa *user agent* telah menerima pesan terakhir dari seangkaian pesan *INVITE*.

BYE yaitu proses terminasi sesi.

CANCEL yaitu proses membatalkan *INVITE*

REGISTER yaitu proses registrasi akun pada *server*

OPTION yaitu proses memento informasi mengenai kemampuan *server*.

INFO Digunakan untuk membawa pesan informasi lainnya, seperti DTMF.

- SIP Response 1.

1xx - *information Message*

2xx - *Successful Response*

3xx - *Redirection Response*

4xx - *Request Failure Response*

5xx - *Server Failure Response*

6xx - *Global Failure Response*

- **Framework SIP.** Pemilihan protokol SIP sebagai protokol *default* dari jaringan IMS karena kemampuan dari protokol SIP terdapat pada *framework* protokol SIP antara lain:
 1. *Integration*, kemudahan untuk integrasi dengan protokol lain standar IETF.
 2. *Scalability*, kemampuan menggabungkan komponen SIP secara fisik dalam *server* yang sama atau berbeda topologi secara topologi. Selain itu, distribusi komponen memungkinkan penambahan komponen baru tanpa mempengaruhi jaringan yang sudah ada.
 3. *Simplicity*, Menangani paket SIP relatif mudah dilakukan seperti menangani paket IETF lainnya seperti HTTP dan SMTP.

C. Layanan Voice dan Video Over IP dengan RTP dan RTCP

Real-Time Transport Protocol (RTP) dan *Real-Time Control Protocol* (RTCP) sebagaimana yang telah dijelaskan dalam RFC 3550, merupakan protokol yang digunakan untuk pengangkutan media stream sejak dahulu alat *conferencing* dibuat. Visual Audio Tool (VAT) menggunakan RTP versi 0. Sejak itu RTP berkembang hingga versi dua. RTP versi 2 tidak kompetibel dengan versi 1, sehingga semua aplikasi harus dibangun untuk mendukung RTP versi dua. Ketika sebuah jaringan menggunakan *statistical multiplexing* untuk mengirimkan data yang *real-time* seperti suara, *jitter* harus diperhitungkan oleh penerima. *Router* adalah sebuah contoh dari perangkat *static multiplexing*, sehingga *voice* dan *video over ip* harus menghadapi masalah *jitter*. RTP didesain untuk memungkinkan penerima untuk memungkinkan penerima untuk mengkompensasi *jitter*. RTP bisa digunakan untuk setiap data yang bersifat *real-time*. RTP mendefinisikan cara memformat muatan paket IP yang membawa data *real-time*.

- RTP (*Real-time Transfer Protocol*)

RTP memungkinkan transportasi data *isochronous* di jaringan paket, yang memperkenalkan *jitter*. *Isynchronous* adalah data yang harus disampaikan secara langsung ketika data tersebut ditangkap. Contoh dari data *isochronous* adalah suara. Contoh lain adalah *video* tetapi data *video* masih mempunyai nilai toleransi yang lebih tinggi. RTP biasanya digunakan diatas protokol UDP. UDP hanya dapat menjamin integritas data dengan menggunakan *checksum*, tetapi aplikasi yang menggunakan protokol UDP harus mengurus segala tugas pemulihan data. UDP juga menyediakan gagasan tentang *port*, dimulai dari 0 dan 65.535 sehingga target dari UDP harus dibedakan antara pengirim dan penerima. Gambar 3 merupakan contoh pengiriman paket UDP dari IP 10.10.10.10:2100 ke 10.10.10.20:3200:

Source IP address:	Source port:	Destination IP address:	Destination port:	RTP Data
10.10.10.10	2100	10.10.10.20	3200	

Gambar 3 Proses Pengiriman Paket

Terlihat pada Gambar 3 bahwa *port* yang digunakan oleh sumber berbeda dengan *port* tujuan.

- RTCP (*Real-time Control Protocol*)

RTCP digunakan untuk mengirimkan paket kontrol ke user mengenai sesi RTP tertentu. Paket ini meliputi berbagai kontrol statistik, informasi tentang *user*, dan informasi mengenai pemetaan peserta untuk individu sumber *stream*. Informasi yang paling berguna ditemukan dalam paket RTCP adalah kualitas transmisi dalam jaringan. Semua *user* dalam sesi yang mengirimkan paket RTCP adalah *sender* sebagai pengirim *report* dan *receiver* sebagai penerima *report*. [3]

D. QoS (*Quality of Service*)

QoS didefinisikan sebagai suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. Untuk menentukan kualitas dari suatu jaringan diperlukan beberapa parameter,

Berikut adalah parameter-parameter yang ada pada QoS:

1. Rate Rasio jumlah bits yang dipindahkan antar dua perangkat dalam satuan waktu tertentu, umumnya dalam detik. Bit rate sama dengan istilah lain *data rate*, *data transfer* dan *bit time*.
2. *Latency* didefinisikan sebagai total waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu titik ke titik lain yang menjadi tujuannya
3. *Packet Loss* atau *Error Packet loss* adalah merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Salah satu penyebab paket *loss* adalah antrian yang melebihi kapasitas *buffer* pada setiap *node*. Beberapa penyebab terjadinya paket *loss* yaitu:
 - a. *Congestion*, disebabkan terjadinya antrian yang berlebihan dalam jaringan
 - b. Node yang bekerja melebihi kapasitas *buffer*
 - c. Memori yang terbatas pada node
 - d. *Policing* atau *control* terhadap jaringan
4. *Jitter* didefinisikan sebagai variasi dari *delay* atau variasi waktu kedatangan paket. Banyak hal yang dapat menyebabkan *jitter*, diantaranya adalah peningkatan trafik secara tiba-tiba sehingga menyebabkan penyempitan *bandwidth* dan menimbulkan antrian. Selain itu, kecepatan terima dan kirim paket dari setiap node juga dapat menyebabkan *jitter*.
5. *Throughput*, yaitu kecepatan (*rate*) *transfer* data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut (sama dengan, jumlah pengiriman paket IP sukses per *service-second*). Persamaan (1) merupakan perhitungan rumus dalam mencari nilai *throughput*:

$$throughput = \frac{\text{paket diterima}}{\text{waktu pengiriman data}} \dots\dots\dots [1]$$

III. ANALISIS DAN PERANCANGAN

A. Analisis dan Kebutuhan Sistem

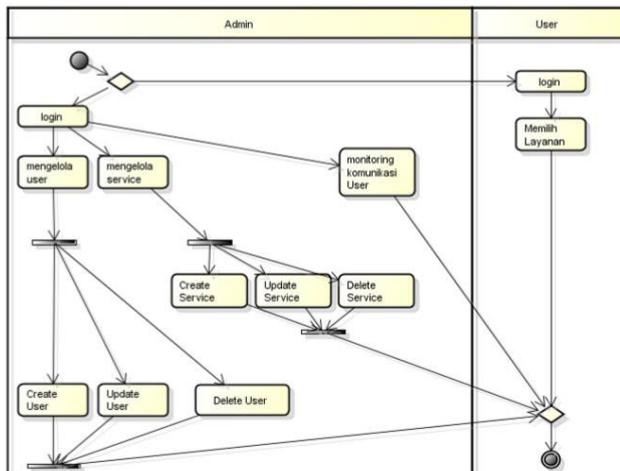
Berikut ini adalah kriteria yang akan dibangun untuk administrator:

1. Administrator dapat melakukan perubahan konfigurasi terhadap *server Open IMS Core*
2. Administrator dapat menambahkan dan menghapus daftar *client* yang terdapat pada HSS *server*
3. Administrator dapat melihat memonitor setiap *user* yang masuk ke dalam *server*
4. Administrator dapat melihat setiap panggilan antar *client*

Sedangkan kriteria yang akan dibangun untuk *client*:

1. *Client* dapat menggunakan *smartphone android* dan *laptop* sebagai media komunikasi
2. *Client* melakukan konfigurasi pada aplikasi *client*, untuk menentukan IP *server IMS* serta *port* untuk melakukan akses ke PCSCF *server*.
3. *Client* memasukkan data identitas di aplikasi *client* sesuai dengan apa yang didaftarkan pada HSS *server*
4. *Client* dapat memilih layanan yang disediakan oleh aplikasi seperti *voice call*, *video call*, atau *messenger*
5. *Client* dapat memonitor trafik jaringan yang terjadi selama komunikasi berlangsung.

Gambar 4 menunjukkan diagram aktivitas administrator dan *user client*.



Gambar 4. Diagram Aktivitas

B. Kebutuhan Perangkat Keras dan Perangkat Lunak

- Kebutuhan perangkat keras jaringan.
Nama komponen : *Access Point TP-LINK TL-WA500G* untuk digunakan sebagai distribusi IP *address* dan media *transfer* data antara *client* dan *server*
- Spesifikasi perangkat *client* dan *Server IMS* (Tabel Rincian kebutuhan perangkat keras dan lunak ditunjukkan pada Tabel I-IV).

TABLE I. SPESIFIKASI SERVER IMS

Hardware	Keterangan
Processor	Intel Core 2 Duo
Clock-Rate	2.1 GHz
Memory RAM	2084 MB DDR2
Harddisk	SATA 160 GB
Ethernet	802.3
WLAN	Wi-fi 802.11 b/g/n

TABLE II. SPESIFIKASI CLIENT IMS (SMARTPHONE)

Hardware	Keterangan
Perangkat keras	Smartphone android : Samsung Galaxy W
Sistem Operasi	Android OS v 2.3 (<i>Ginger bread</i>)
WLAN	Wi-fi 802.11 b/g/n

TABLE III. SPESIFIKASI CLIENT IMS (LAPTOP)

Hardware	Keterangan
Perangkat keras	Laptop
Sistem Operasi	Windows 7
WLAN	Wi-fi 802.11 b/g/n
Ethernet	802.3

- Spesifikasi Perangkat Lunak

TABLE IV. SPESIFIKASI PERANGKAT LUNAK

Perangkat Lunak	Spesifikasi	Keterangan
Ubuntu	Ubuntu 10.04	Sistem Operasi yang digunakan sebagai <i>server</i> dari <i>Open IMS core</i>
	DNS	Digunakan untuk menterjemahkan alamat IP ke alamat nama
Open IMS Core	Ser_ims dan FHoSS	Aplikasi yang digunakan sebagai <i>server</i> dari IMS serta <i>server database client</i> IMS
Wireshark	Network Analyzer	Aplikasi yang digunakan untuk analisa <i>protocol</i> dan menghitung kualitas layanan
IMS Droid	Aplikasi <i>client</i> pada <i>android</i>	Aplikasi yang digunakan untuk melakukan login pada <i>server</i> , kemudian memilih layanan multimedia yang diinginkan pada sistem operasi <i>android</i>
Boghe	Aplikasi <i>client</i> pada sistem operasi <i>windows</i>	Aplikasi yang digunakan untuk melakukan login pada <i>server</i> , kemudian memilih layanan multimedia yang diinginkan sistem operasi <i>windows</i> .

- Kebutuhan IP Interface ditunjukkan pada Tabel 5.

TABLE V. KEBUTUHAN IP INTERFACE

Perangkat	Net ID	Interface	Range	Keterangan
Server	192.168.1.0/24	WLAN	-	IP Fix yang didapat dari access point
Client	192.168.1.0/24	WLAN	192.168.1.2 – 192.168.1.254	IP yang didapat dari DHCP access point

C. Arsitektur dan Rancangan Topologi

Pada desain jaringan ini server-server IMS dikumpulkan pada satu laptop. IMS server dihubungkan ke sebuah access point menggunakan kabel UTP. Untuk melakukan koneksi client harus terhubung ke access point kemudian mendapatkan IP DHCP yang disebarkan melalui access point.

Gambar 5 merupakan topologi yang digunakan dalam penelitian:



Gambar 5. Topologi Jaringan pada Perancangan

IV. IMPLEMENTASI DAN PENGUJIAN

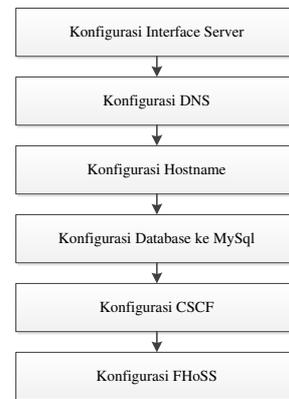
A. Implementasi

Implementasi arsitektur IMS menggunakan Open IMS Core dilakukan pada sistem operasi Ubuntu 10.04. Langkah konfigurasi Open IMS Core ditunjukkan pada Gambar 6. Langkah-langkah implementasi yang dilakukan adalah sebagai berikut:

- Compile ser_ims dan Compile FHoSS
Lakukan mengcompile ser_ims dan FHoSS. Jika terjadi error pada saat compile FHoSS, tentukan target Java homonya.
Langkah untuk menentukan target java home adalah

```
root@alto:/#export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

- Konfigurasi Open IMS Core
Berikut langkah-langkah konfigurasi Open IMS Core:



Gambar 6. Konfigurasi Open IMS Core

Konfigurasi Interface Server

```
root@alto:/#nano/etc/network/interfaces
```

Konfigurasi DNS

```
root@alto:/#cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone/etc/bind/rafkiIMS.dnszone
root@alto:/#nano /etc/bind/rafkiIMS.dnszone
root@alto:/#nano /etc/bind/rafkiIMS.rev
root@alto:/#nano /etc/bind/named.local.conf
```

Konfigurasi Hostname

```
root@alto:/#nano /etc/hosts
```

Konfigurasi Database ke MySql

```
root@alto:/# mysql -u root -p -h localhost </opt/OpenIMSCore/ser_ims/cfg/icscf.sql
root@alto:/# mysql -u root -p -h localhost </opt/OpenIMSCore/FHoSS/scripts/hss_db.sql
root@alto:/# mysql -u root -p -h localhost </opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

Konfigurasi server CSCF

Berikut adalah langkah-langkah konfigurasi CSCF:

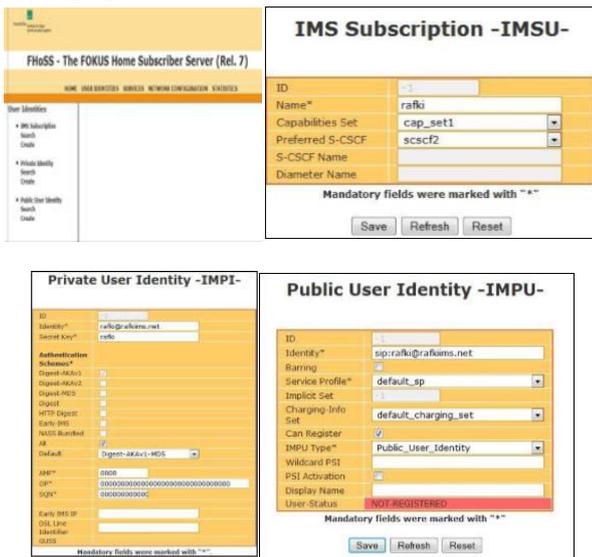
```
root@alto:/#cd /opt/OpenIMSCore
root@alto:/opt/OpenIMSCore/#cp /ser_ims/cfg/*.sh /opt/OpenIMSCore
root@alto:/opt/OpenIMSCore/#cp /ser_ims/cfg/*.cfg /opt/OpenIMSCore
root@alto:/opt/OpenIMSCore/#cp /ser_ims/cfg/*.xml /opt/OpenIMSCore
```

Langkah pertama sebelum melakukan konfigurasi server CSCF adalah, memindahkan semua konfigurasi berekstensi .sh, .cf, dan .xml yang berada pada "/ser_ims/cfg" ke direktori "/opt/OpenIMSCore."

Konfigurasi FHoSS

Setelah melakukan konfigurasi pada tiap-tiap *server* CSCF, Langkah selanjutnya adalah mendaftarkan *user* sesuai dengan domain yang telah dibuat pada HSS *server*. Pada saat mengeksekusi file `./fhoSS` biasanya akan terjadi *error*, hal ini disebabkan file `startup.sh` yang berada pada `~/opt/OpenIMSCore/FHoSS/deploy/` belum diedit java homeny. Caranya adalah menambahkan target javanya, kemudian konfigurasi HSS untuk menambahkan user pada server IMS. Akses `rafkiims.net:8080` untuk *login* dengan menggunakan *username* `hssAdmin` dengan *password* `hss`.

Gambar 7 menunjukkan konfigurasi FHoSS pada server.



Gambar 7. Konfigurasi FHoSS pada Server

- Instalasi Aplikasi *Client*.
Konfigurasi *IMS Client*.
Setelah mengunduh paket *IMDroid* untuk *smartphone android* dan *boghe* untuk aplikasi IMS pada *laptop* langkah selanjutnya adalah masuk ke menu option untuk konfigurasi pada aplikasi *client*. Hal yang harus dilakukan adalah menambahkan identitas yang sesuai dengan apa yang telah didaftarkan pada *Home Subscriber Server* (HSS). Selanjutnya lakukan konfigurasi pada menu *network*. Isi IP *address* dari *server* kemudian *port* dari *server*. Terakhir, lakukan *sign in*.

B. Pengujian

Pengujian dilakukan dengan beberapa skenario. Pengujian dan hasil pengujian dapat dilihat pada uraian berikut :

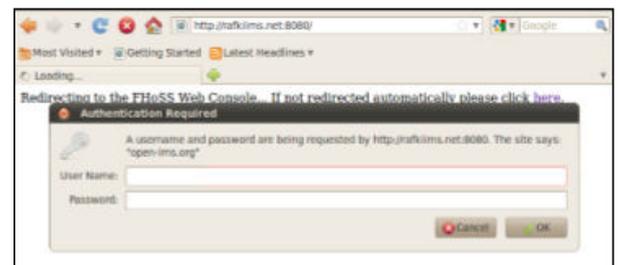
- Pengujian DNS dilakukan untuk menterjemahkan domain server IMS dari alamat IP ke alamat nama. Parameter yang diukur

adalah Alamat IP dan alamat nama dengan perintah : `nslookup`. Pengujian ini berhasil dilakukan dengan penterjemahan IP `192.168.1.1` yang merupakan server DNS dari domain `rafkiims.net`.

```
Capture :root@alto:#nslookup
Server: 192.168.1.1
Address: 192.168.1.1#53
1.1.168.192.in-addr.arpa
name = rafkiims.net.
```

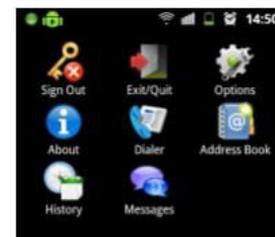
- Pengujian FHoSS
Pengujian FHoSS dilakukan untuk menguji database pengguna layanan IMS. Parameter yang diukur adalah FHoSS dengan perintah : `./fhoSS`. FHoSS berhasil diuji pada saat mengakses `rafkiims.net:8080` pada web *interface* muncul autentikasi *user*. Hasil *capture* pengujian FHoSS ditunjukkan pada Gambar 8.

```
root@alto-laptop:/opt/OpenIMSCore# ./fhoSS.sh
Building Classpath
Classpath is lib/xml-apis.jar:lib/xercesImpl.jar:lib/xerces-2.4.0.jar:lib/xalan-2.4.0.jar:lib/toncat-util.jar:lib/toncat-http.jar:lib/toncat-coyote.jar:lib/struts.jar:lib/servlets-default.jar:lib/servlet-api.jar:lib/naming-resources.jar:lib/naming-factory.jar:lib/mysql-connector-java-3.1.12-bin.jar:lib/mx4j-3.0.1.jar:lib/log4j.jar:lib/junit.jar:lib/junit4.jar:lib/jta.jar:lib/jsp-api.jar:lib/jmx.jar:lib/jsp.jar:lib/jasper-runtime.jar:lib/jasper-compiler-jdt.jar:lib/jasper-compiler.jar:lib/hibernate.jar:lib/FHoSS.jar:lib/ehcache-1.1.jar:lib/dom4j-1.6.1.jar:lib/commons-validator.jar:lib/commons-modeler.jar:lib/commons-logging.jar:lib/commons-logging-1.0.4.jar:lib/commons-lang.jar:lib/commons-fileupload.jar:lib/commons-el.jar:lib/commons-digester.jar:lib/commons-collections-3.1.jar:lib/commons-beanutils.jar:lib/rglib-2.1.3.jar:lib/catalina-optional.jar:lib/catalina.jar:lib/C3P0-0.9.1.jar:lib/base64.jar:lib/asa.jar:lib/asa-attrs.jar:lib/antir-2.7.6.jar:log4j.properties:..
2012-06-22 14:43:52,463 INFO de.fhg.fokus.hss.main.TomcatServer - startTomcat
Tomcat Server is started.
2012-06-22 14:43:54,023 WARN org.apache.catalina.connector.MapperListener - registerEngine Unknown default host: 192.168.1.1
2012-06-22 14:43:55,265 INFO de.fhg.fokus.hss.web.servlet.ResponseFilter - init
Response Filter Initialisation!
2012-06-22 14:43:55,886 INFO de.fhg.fokus.hss.main.TomcatServer - startTomcat
WebConsole of FHoSS was started!
```



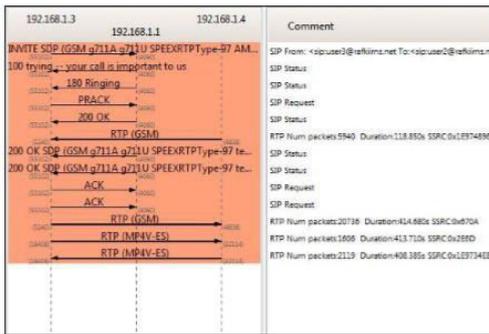
Gambar 8. Login FHoSS

- Pengujian *Client*
Digunakan untuk memilih dan menggunakan layanan multimedia. Parameter yang diukur adalah *login*. Dengan memilih perintah *sign in*, aplikasi berhasil menampilkan menu (Gambar 9) yang disediakan untuk pengguna IMS agar dapat saling berkomunikasi maupun menggunakan layanan multimedia lainnya.



Gambar 9. Multimedia IMSDroid.

- Pengujian Protokol
Gambar 10 merupakan hasil pengujian protokol SIP pada saat melakukan panggilan suara dan *video*.



Gambar 10 Komunikasi *Client* menggunakan protokol SIP

- Pengujian *VoiceCall*
Digunakan untuk melakukan panggilan suara. Dengan memanggil identitas *user* yang dituju dan memilih menu *voice call*.
Contoh : user1@rafkiims.net. Kemudian terjadi proses komunikasi suara antar *client* (Gambar 11).



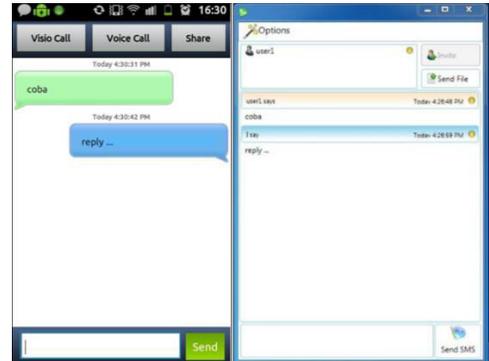
Gambar 11. *Voice Call* pada *IMSDroid* dan *Boghe*

- Pengujian *Video Call*.
Digunakan untuk melakukan panggilan video bersuara. Dengan memanggil identitas *user* yang dituju dan memilih menu *video call*. Contoh: user1@rafkiims.net, maka terjadi proses komunikasi *video call* antar *client* (Gambar 12).



Gambar 12. *Video Call* pada *IMSDroid* dan *Boghe*

- Pengujian *Messaging*
Pengujian (Gambar 13) ini bertujuan untuk menguji fitur untuk melakukan komunikasi dengan teks. Parameter yang diukur adalah pengiriman pesan antar *client*. Dengan memanggil identitas *user* yang dituju dan memilih menu *instant messaging*: user1@rafkiims.net



Gambar 13. Pengujian *Messaging* pada *IMSDroid* dan *Boghe*

- Pengujian *File Sharing*



Gambar 14. Pengujian *File sharing* pada *IMSDroid* dan *Boghe*

Pengujian (Gambar14) ini dilakukan dengan tujuan pengujian untuk berbagi *file* antar *client*. Parameter yang diukur adalah pengiriman *file* antar *client*, dengan memanggil identitas *user* yang dituju dan memilih menu *sharing file*: user1@rafkiims.net. Hasil yang diharapkan adalah adanya proses *sharing file* antar *client*.

- Pengujian QoS
Pengujian QoS dilakukan di lobby Fakultas Ilmu Terapan (FIT) dengan *access point* berada di bagian tengah *lobby*. 1 buah *Client* berada di Lab Bahasa dan 2 *client* lainnya berada di lobby FIT.

TABLE VI. PENGUJIAN DENGAN 2 *CLIENT*

Waktu capture	Payload	Asal	Tujuan	delay (ms)
5 menit	GSM	192.168.1.2	192.168.1.3	10.225
	MP4V-ES	192.168.1.2	192.168.1.3	50.261
	GSM	192.168.1.3	192.168.1.2	18.222
	MP4V-ES	192.168.1.3	192.168.1.2	117.86

Berdasarkan Tabel VI dapat disimpulkan bahwa komunikasi *voice (GSM)* dan *video call (MP4V-ES)* terjadi. Pengambilan trafik selama 5 menit memperlihatkan delay dalam kategori *excellent* karena berada pada range < 150 ms.

TABLE VII. PENGUJIAN DENGAN 4 CLIENT

Waktu capture	Payload	Asal	Tujuan	Delay (ms)
5 menit	GSM	192.168.1.4	192.168.1.6	18.155
	MP4V-ES	192.168.1.4	192.168.1.6	64.595
	GSM	192.168.1.6	192.168.1.4	46.941
	MP4V-ES	192.168.1.6	192.168.1.4	65.977
	GSM	192.168.1.5	192.168.1.7	14.962
	H264	192.168.1.5	192.168.1.7	256.161
	GSM	192.168.1.7	192.168.1.5	22.395
	H264	192.168.1.7	192.168.1.5	59.946

Pengujian kualitas layanan pada Tabel VII dilakukan menggunakan 4 *client* dan waktu pengambilan selama 5 menit. Pengambilan trafik jaringan dilakukan pada saat semua *client* melakukan panggilan suara dan *video* secara bersama-sama. Berdasarkan kondisi di atas maka dapat disimpulkan bahwa *delay* yang didapatkan masih berada dalam kategori *excellent* untuk beberapa komunikasi dan kategori *good* untuk komunikasi H264, karena pada jangkauan 150 ms < *delay* < 300 ms.

Tabel VIII-IX merupakan hasil *capture* paket untuk pengujian *jitter* (variasi kedatangan paket) *video call* dengan pengujian 2 *client* dan 4 *client*:

TABLE VIII. PENGUJIAN JITTER DENGAN 2 CLIENT

Waktu capture	Payload	Asal	Tujuan	Jitter (ms)
5 menit	GSM	192.168.1.2	192.168.1.3	29.3
	MP4V-ES	192.168.1.2	192.168.1.3	0.05
	GSM	192.168.1.3	192.168.1.2	2.91
	MP4V-ES	192.168.1.3	192.168.1.2	3.65

TABLE IX. PENGUJIAN JITTER DENGAN 4 CLIENT

Waktu capture	Payload	Asal	Tujuan	Jitter (ms)
5 menit	GSM	192.168.1.4	192.168.1.6	5.63
	MP4V-ES	192.168.1.4	192.168.1.6	2.45
	GSM	192.168.1.6	192.168.1.4	60.61
	MP4V-ES	192.168.1.6	192.168.1.4	0
	GSM	192.168.1.5	192.168.1.7	13.56
	H264	192.168.1.5	192.168.1.7	50.31
	GSM	192.168.1.7	192.168.1.5	13.29
	H264	192.168.1.7	192.168.1.5	38.73

Tabel VIII-IX menunjukkan bahwa nilai *jitter* pada komunikasi 2 dan 4 *client* yang saling melakukan panggilan suara dan video dengan *payload* GSM untuk panggilan suara, MP4V-ES dan H264 untuk panggilan video masuk pada kategori baik karena berada pada range 0 s/d 75 ms.

Tabel 10-11 merupakan hasil *capture* paket untuk pengujian *video call* dan *voice call* untuk parameter *throughput* dengan pengujian 2 dan 4 *client*:

TABLE X. PENGUJIAN THROUGHPUT DENGAN 2 CLIENT

Waktu capture	Payload	Asal	Tujuan	Throughput (Kbps)
5 menit	GSM	192.168.1.2	192.168.1.3	11.972
	MP4V-ES	192.168.1.2	192.168.1.3	147.369
	GSM	192.168.1.3	192.168.1.2	22.776
	MP4V-ES	192.168.1.3	192.168.1.2	61.858

TABLE XI. PENGUJIAN THROUGHPUT DENGAN 4 CLIENT

Waktu capture	Payload	Asal	Tujuan	Throughput (Kbps)
5 menit	GSM	192.168.1.4	192.168.1.6	22.776
	MP4V-ES	192.168.1.4	192.168.1.6	79.211
	GSM	192.168.1.6	192.168.1.4	17.32
	MP4V-ES	192.168.1.6	192.168.1.4	214.339
	GSM	192.168.1.5	192.168.1.7	21.227
	H264	192.168.1.5	192.168.1.7	116.582
	GSM	192.168.1.7	192.168.1.5	24.292
	H264	192.168.1.7	192.168.1.5	248.867

Throughput merupakan total kedatangan paket yang dapat diamati pada suatu *interval* waktu tertentu dan juga merupakan kemampuan dari jaringan dalam melakukan pengiriman data. Berdasarkan Tabel X-XI *throughput* yang didapatkan berada pada kondisi yang cukup baik, yaitu masih di bawah 250Kbps.

V. KESIMPULAN

Berdasarkan hasil pengujian dari penelitian ini, maka dapat disimpulkan bahwa:

1. Arsitektur IMS dapat berjalan pada jaringan *intranet*, karena *server* CSCF seperti P-CSCF, S-CSCF, dan I-CSCF dapat berjalan tanpa adanya error, kemudian pada hasil pengujian tiap *server* CSCF memberikan laporan "OPEN", hal ini menyatakan bahwa *server* CSCF siap untuk melayani permintaan dari *user*.
2. Implementasi protokol SIP (*Session Initiation Protocol*) pada arsitektur IMS (*IP Multimedia Subsystem*) berjalan dengan baik, karena layanan-layanan yang dapat ditangani oleh protokol SIP seperti *voice call*, *video call*, *instant messaging*, dan *file sharing* dapat berjalan dengan baik.
3. Layanan komunikasi dan multimedia dapat dinikmati dari berbagai perangkat seperti *smartphone* berbasis *android*, *laptop* dan PC sesuai dengan hasil pengujian.
4. Berdasarkan hasil pengujian, analisis QoS yang didapatkan berada pada kondisi yang baik, yaitu untuk *delay* berada pada kondisi *excellent*, *jitter* pada kondisi baik yaitu berada pada range 0 s/d 75ms dan nilai *throughput* berada di kisaran 250 Kbps.

REFERENCES

- [1] T.C Henning dkk, "Kualitas Layanan IP Multimedia Subsystem", ITS Surabaya, Agustus 2010
- [2] Ismail, nanang dkk, "IP MULTIMEDIA SUBSYSTEM (IMS) MENDORONG MUNCULNYA PELUANG DAN MODEL BISNIS BARU, STEI ITB", May 2006.
- [3] Kevin Purtell, Dhruv Mohindra, Wallace T. Clark IV John Bennett, "IP Multimedia Subsystem," Final Report, pp. 5-11, 2007.

- [4] Triko Marcello, IMPLEMENTASI DAN ANALISIS JARINGAN VOIP PADA JARINGAN INTERNAL POLITEKNIK TELKOM. Bandung: Politeknik Telkom, 2010.
- [5] Olivier Hersent, IP Telephony Deploying VoIP Protocol and IMS Infrastructure, second Edition. India: Wiley, 2011.
- [6] Camarillo, Gonzalo and Miguel A. Garcia Martin. 2004. The 3G IP Multimedia Subsystem (IMS). Finland.