



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI ENUM SERVER PADA JARINGAN IMS  
DENGAN MENGGUNAKAN OPEN IMS CORE**

**SKRIPSI**

FAISAL JAMIL

0706267673

DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA

DEPOK

JUNI 2011



**UNIVERSITAS INDONESIA**

**IMPLEMENTASI ENUM SERVER PADA JARINGAN IMS  
DENGAN MENGGUNAKAN OPEN IMS CORE**

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

FAISAL JAMIL

0706267673

DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA

DEPOK

JUNI 2011

## HALAMAN PERNYATAAN ORISINALITAS

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

NAMA : Faisal Jamil

NPM : 0706267673

Tanda Tangan :



Tanggal : 15 JUNI 2011

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Faisal Jamil  
NPM : 0706267673  
Program Studi : Teknik Elektro  
Judul Skripsi : Implementasi ENUM Server pada Jaringan IMS  
dengan Menggunakan Open IMS Core

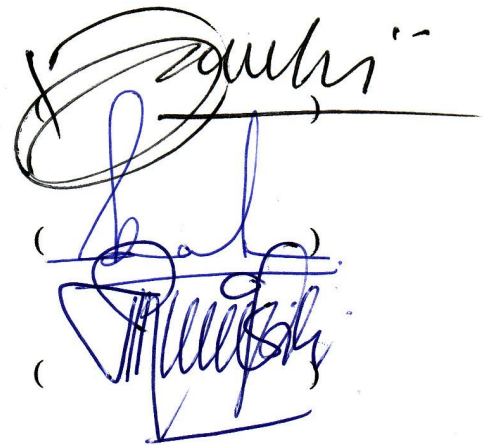
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

## DEWAN PENGUJI

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

Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng

Penguji : Ir. Endang Sriningsih MT, Si



Ditetapkan di : Depok

Tanggal : 27 Juni 2011

## KATA PENGANTAR

Segala puji yang terbaik, saya sampaikan kepada Allah SWT atas limpahan nikmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Sesungguhnya saya sangat menyadari kelemahan diri saya, dan Maha Besar Allah yang berkenan memudahkan penulisan skripsi ini. Penulisan skripsi ini dilakukan sebagai salah satu syarat untuk menjadi Sarjana Teknik di Departemen Teknik Elektro FTUI. Saya pun menyadari bahwa tanpa bantuan banyak pihak, skripsi ini akan sulit terselesaikan. Oleh sebab itu, pada kesempatan ini saya ingin mengucapkan terima kasih kepada pihak-pihak yang membantu saya dalam segala hal mengenai penyusunan skripsi ini yaitu:

- 1) Ir. Djamhari Sirat M.Sc, Ph.D selaku dosen Pembimbing seminar dan skripsi yang telah meluangkan waktunya serta masukan-masukan selama bimbingan.
- 2) Kedua orang tua tercinta : Ibu Hj.Ruhaesti, Bapak H.M.Maula, dan kakak Farah Zakia Ahdiana yang telah memberikan dukungan doa yang sangat berarti.
- 3) Dipl.-Ing. Franz Elder dari OpenIMSCore Developer, yang telah mengajari saya banyak hal tentang IMS dan pemecahan masalahnya.
- 4) Mas Tony Suryo dan Bapak Agus Awaludin, para trainer Huawei Training Center yang telah memberikan ilmunya dan kesempatannya untuk saya mempelajari perangkat Huawei.
- 5) Rekan-rekan satu bimbingan: Rosa, Chandra Gunawan, Ardy Thiotrisno, Krisna Juanta yang membantu dan bekerja sama dalam skripsi ini.
- 6) Rekan-rekan asisten Laboratorium Telekomunikasi, asisten Laboratorium Jaringan, Rekan-rekan guru & siswa CK-12, Sospol UI, dan Komunitas Saung Ilmu tercinta.
- 7) Rekan-rekan Elektro angkatan 2007 dan seluruh civitas akademik Departemen Teknik Elektro

Depok, Juni 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI  
UNTUK KEPENTINGAN AKADEMIS**

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

Nama : Faisal Jamil  
NPM : 0706267673  
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 Royalty Free*)** atas karya ilmiah saya yang berjudul :

**IMPLEMENTASI ENUM SERVER PADA JARINGAN IMS  
DENGAN MENGGUNAKAN OPEN IMS CORE**

Beserta perangkat yang ada (Jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis dan pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 15 Juni 2011

Yang menyatakan



(Faisal Jamil)

## ABSTRAK

Nama : Faisal Jamil  
Program Studi : Teknik Elektro  
Judul : Implementasi ENUM Server Pada Jaringan IMS , Dengan Menggunakan Open IMS Core

Skripsi ini membahas tentang implementasi ENUM Server pada jaringan IMS. ENUM server adalah sebuah solusi penomoran dalam terciptanya hubungan antara jaringan yang menggunakan sistem penomoran konvensional dan jaringan yang menggunakan identitas URI dalam jaringan Internet. Dengan implementasi ENUM Server pada jaringan IMS, pengguna IMS dapat lebih fleksibel dalam melakukan koneksi ke siapapun tujuannya. Tetapi implementasi ENUM Server dalam jaringan IMS akan mempengaruhi tahapan yang terjadi dalam melakukan panggilan. Tahapan ini akan mempengaruhi waktu proses saat melakukan panggilan. Dari data percobaan didapatkan waktu proses pada panggilan yang tidak melalui ENUM Server sebesar 0.1071274 detik (107.2 ms) , sementara waktu proses pada panggilan yang melalui ENUM Server sebesar 0.1332041 detik (133.2 ms). Jika dibandingkan dengan standard waktu respon VoIP yang ada yaitu maksimum sebesar 205-218 ms, maka dapat disimpulkan bahwa baik waktu proses panggilan yang melalui ENUM Server maupun yang tidak melalui ENUM Server masih memenuhi standard yang ada.

Kata Kunci : ENUM , IMS, Tahapan pemanggilan, Waktu Proses

## ABSTRACT

Name : Faisal Jamil  
Study Program : Electrical Engineering  
Title : Implementation of ENUM Server on IMS Network, using Open IMS Core

This thesis discussed about the implementation of ENUM server on IMS Network. ENUM server is a solution for numbering process between the conventional numbering system and URI identity-based networks. By implementing ENUM server on IMS network, IMS user could improve their connection flexibility on any numbering identity. However the implementation of ENUM Server to IMS network will effect the steps on calling process. Eventually this will lead to an addition of time on the calling process. From the simulation, a 0,1071274 seconds is obtained by the calling process without ENUM Server, While a 0,1332041 seconds is acquired by the calling process using ENUM Server. When compared with the maximum standard response time of existing VoIP (maximum 205-218 ms) , it can be concluded that both the processing time of the call through ENUM Server or not via ENUM Server still meet existing standards.

Keywords : ENUM, IMS, Calling Steps, Process Time



## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Pembatasan Masalah.....	3
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	3
BAB 2 IMS, TEL URI DAN ENUM.....	5
2.1. Jaringan IMS.....	5
2.1.1. Komponen-komponen IMS.....	6
2.2. Protokol-protokol yang Digunakan Dalam IMS.....	10
2.2.1. SIP Protocol.....	10
2.2.2. SDP.....	16
2.2.3. Diameter.....	18
2.3. TEL URI.....	19
2.4. <i>E.164 Number Mapping</i> (ENUM).....	21
2.4.1. Aplikasi ENUM.....	22
2.4.2. Konversi format E.164 ke ENUM.....	23
2.4.3. ENUM NAPTR DATA.....	23
2.5. Lama Waktu dalam Respon Pencarian.....	25

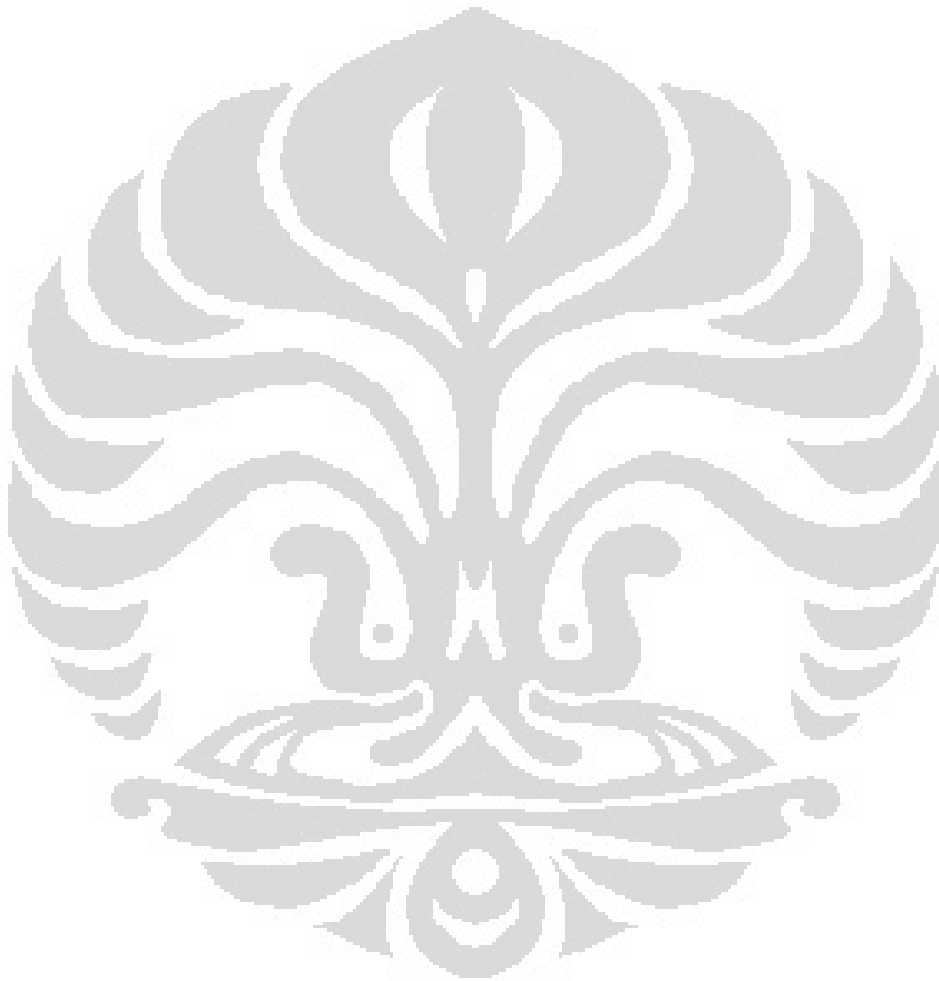
BAB 3	SKENARIO PENGGUNAAN ENUM SERVER PADA OPEN IMS CORE.....	26
3.1.	Skenario Penggunaan ENUM Server dalam Jaringan IMS.....	26
3.2.	Penggunaan Modul ENUM dalam OpenIMSCore.....	28
3.2.1.	Fungsi enum_query.....	29
3.2.2.	Proses Load Modul ENUM dalam OpenIMSCore .....	31
3.3.	Konfigurasi ICSCF dan SCSCF .....	31
3.4.	Konfigurasi ENUM Server.....	32
3.5.	Pengujian Konfigurasi ENUM Server dari Komputer Server.....	33
3.6.	Pengujian Konfigurasi ENUM Server dari Komputer <i>Client</i> .....	34
BAB 4	HASIL PENGUJIAN DAN ANALISIS .....	37
4.1.	Prosedur Pengambilan Data .....	37
4.2.	Topologi Jaringan Percobaan .....	38
4.3.	Proses Kerja Panggilan Menggunakan SIP URI .....	39
4.4.	Proses Kerja Panggilan Menggunakan TEL URI.....	42
4.5.	Perbandingan Lama Waktu Proses Kerja .....	45
BAB 5	KESIMPULAN .....	48
DAFTAR ACUAN	.....	49

## DAFTAR GAMBAR

Gambar 2.1 Konsep Integrasi Horizontal Layanan [5] .....	6
Gambar 2.2 Jaringan IMS [6].....	7
Gambar 2.3 SIP server sebagai <i>Registrar Server</i> [7] .....	12
Gambar 2.4 SIP server sebagai <i>Proxy Server</i> [7].....	12
Gambar 2.5 SIP server sebagai <i>Redirect Server</i> [7].....	13
Gambar 2.6 . Komponen CSCF Akses Database ENUM Server [12].....	22
Gambar 3.1 Skenario Penggunaan Umum ENUM Server.....	26
Gambar 3.2. Skenario Penggunaan ENUM Server dalam Jaringan IMS .....	28
Gambar 3.3. Contoh Pengaturan Data Pada ENUM Server .....	32
Gambar 3.4. Konfigurasi Logging .....	33
Gambar 3.5 Pengujian Konfigurasi ENUM Server dari Komputer Server.....	34
Gambar 3.6 . Hasil Ping domain e164.arpa dari Terminal Komputer <i>Client</i> .....	35
Gambar 3.7. Pengujian Konfigurasi dengan Ping e164.arpa Menggunakan <i>Wireshark</i> .....	35
Gambar 3.8 Pengujian Konfigurasi ENUM Server dengan Dig Menggunakan <i>Wireshark</i> .....	36
Gambar 4.1. Mercurio Bronze IMS Client.....	38
Gambar 4.2 Topologi Jaringan ENUM Server - OpenIMSCore.....	38
Gambar 4.3 Alur Pesan Pemanggilan Menggunakan SIP URI.....	39
Gambar 4.4 Alur Pesan Pemanggilan Menggunakan tel URI.....	42

## DAFTAR TABEL

Tabel 2.1 Jenis-Jenis Layanan yang di dukung ENUM Server .....	23
Tabel 4.1. Data Lama Waktu Kerja Pemanggilan Format SIP URI .....	45
Tabel 4.2. Data Lama Waktu Kerja Pemanggilan Format TEL URI.....	46
Tabel 4.3 Data Waktu Proses Akses ENUM Server .....	47



# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi telekomunikasi saat ini telah membuat meningkatnya kebutuhan pelanggan akan aplikasi telekomunikasi yang baru dan menarik. Operator dituntut menyediakan berbagai layanan multimedia baru yang menarik bagi pelanggan. Namun disisi lain, operator juga tidak dapat melupakan pelanggan dengan jaringan yang telah digunakan saat ini (*legacy network*). Investasi yang tinggi pada PSTN (*legacy network*) merupakan alasan penting mengapa operator harus mempertahankan pelanggan yang telah ada.

*IP Multimedia Subsystem* (IMS) adalah bagian dari konsep NGN yang berupa arsitektur multimedia untuk layanan IP bergerak dan tetap (*mobile and fixed IP*). IMS merupakan arsitektur multimedia yang terbuka, telah terstandarisasi, dan ramah operator. IMS sangat menjanjikan karena menggabungkan antara dunia Internet dan selular dengan layanan *fixed line* dengan menggunakan teknologi seluler untuk menyediakan akses luas dan teknologi Internet untuk menyediakan berbagai layanan baru yang menarik. IMS juga menjanjikan penurunan biaya untuk *maintenance* infrastruktur dengan hanya menggunakan satu jenis jaringan transport untuk setiap akses jaringan.

Interkoneksi jaringan PSTN dengan jaringan berbasis IMS merupakan salah satu hal yang patut diperhitungkan karena jumlah pengguna jaringan jaringan *fixed line* yang sangat besar merupakan suatu potensi pasar yang harus dipertahankan dalam perkembangan teknologi mendatang.

Sistem penomoran merupakan salah satu faktor kunci yang diperlukan dalam sinergisasi layanan telepon konvensional dengan layanan IMS yang berbasis Internet. [1] Diperlukan suatu penghubung yang terstandarisasi agar IMS dapat berjalan dengan mudah dalam berkomunikasi. [2].

Untuk mengatasi masalah ini, dibutuhkan suatu penghubung standard untuk menjembatani perbedaan ini. ENUM (*E164 Number Mapping*) adalah

solusi yang dapat mentranslasi nomor telepon menjadi URI (*Uniform Resource Identifier*) atau alamat IP yang dapat digunakan dalam komunikasi Internet. [3] Dengan adanya ENUM, komunikasi antara devais dengan format penomoran telepon dengan format IP dapat menjadi lebih sederhana, murah dan tanpa banyak merubah perangkat di sisi pelanggan.

## 1.2. Perumusan Masalah

Dalam skripsi ini akan dilakukan penerapan ENUM dalam jaringan IMS untuk melihat pengaruh penerapan ENUM dalam jaringan IMS. Terdapat beberapa rumusan masalah yang akan diperhatikan dan dianalisis, yaitu: ENUM adalah sebuah teknologi server yang dapat mentranslasi nomor telepon menjadi URI, format standar yang digunakan dalam komunikasi Internet. ENUM diperlukan sebagai “jembatan” antara komunikasi berformat penomoran konvensional dengan IMS yang berbasis IP.

Berdasarkan gambaran diatas, maka bahasan pokok yang akan dibahas pada skripsi ini adalah :

1. Bagaimanakah cara kerja ENUM server dalam mentranslasi format pengenalan telepon dan Internet?
2. Bagaimanakah implementasi ENUM server didalam IMS ?
3. Bagaimanakah tahapan yang terjadi didalam IMS ketika menggunakan ENUM server dalam proses pemanggilan ?
4. Bagaimanakah waktu proses yang terjadi saat dilakukan panggilan melalui ENUM server dan tidak melalui ENUM server?

## 1.3. Tujuan

Tujuan dari penyusunan skripsi ini adalah:

1. Mengimplementasikan cara kerja ENUM server dalam mentranslasi format pengenalan telepon dan internet
2. Membuktikan tahapan yang terjadi didalam IMS ketika menggunakan ENUM server dalam proses pemanggilan
3. Membandingkan waktu proses yang terjadi saat dilakukan panggilan melalui ENUM server dan tidak melalui ENUM server serta membandingkan keduanya dengan standar yang ada.

#### 1.4. Pembatasan Masalah

Pada skripsi ini, masalah dibatasi pada mengetahui tahapan yang terjadi saat panggilan panggilan dengan menggunakan format telepon di dalam jaringan IMS dan perbandingan lama waktu proses yang terjadi antara panggilan yang melalui ENUM Server dan tidak melalui ENUM server.

#### 1.5. Metode Penelitian

Metode penelitian yang digunakan dalam penyusunan skripsi ini adalah:

1. Tinjauan Pustaka, yaitu dengan melakukan studi literatur dari buku-buku pustaka, *paper*, *website*, dan modul-modul yang berkaitan dengan skripsi ini.
2. Diskusi dan tanya jawab dengan dosen pembimbing dan orang-orang yang ahli pada bidang yang berkaitan dengan skripsi ini, seperti pengembang dari Open IMS Core.
3. Simulasi percobaan

#### 1.6. Sistematika Penulisan

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

##### 1. Bab 1 : PENDAHULUAN

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

##### 2. Bab 2 : IMS, TEL URI DAN ENUM

Pada bab ini akan dijelaskan landasan teori tentang *IP Multimedia Subsystem (IMS)*, format TEL URI, konsep dan cara kerja ENUM.

##### 3. Bab 3 : SKENARIO PENGGUNAAN ENUM SERVER PADA OPEN IMS CORE

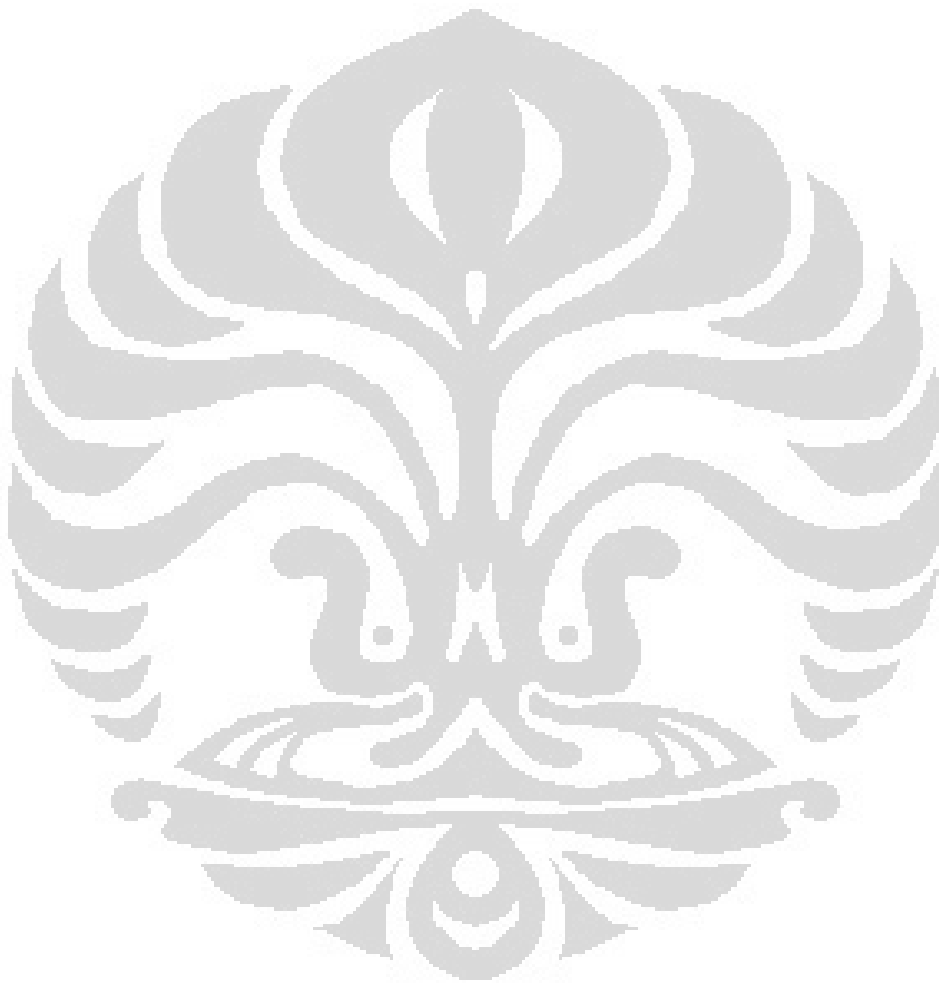
Pada bab ini, akan dijelaskan tentang skenario penggunaan ENUM Server, cara mengkonfigurasi ENUM Server, dan konfigurasi penggunaan ENUM dalam OpenIMScore.

4. Bab 4 : HASIL PENGUJIAN DAN ANALISIS

Pada bab ini pengolahan data dari hasil percobaan dan analisis dari tiap percobaan.

5. Bab 5: KESIMPULAN

Pada bab ini akan ditarik kesimpulan dari hasil pengujian.





## BAB 2

### IMS, TEL URI DAN ENUM

#### 2.1. Jaringan IMS

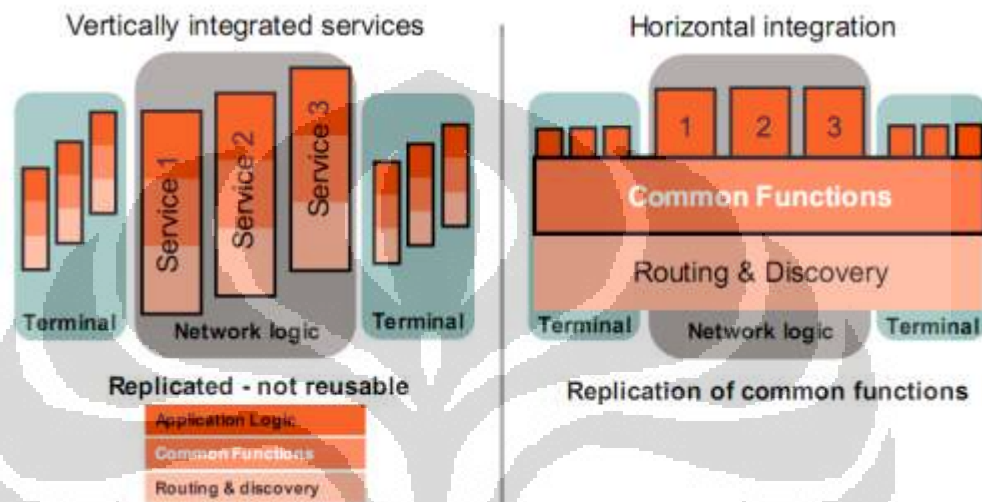
Pada era multimedia saat ini, pengguna layanan telekomunikasi menginginkan dapat menggunakan suatu yang lebih dari layanan komunikasi mereka, dengan harga yang lebih murah, dan menunjukkan suatu yang menarik dari layanan selain suara. Mereka tertarik dengan layanan yang menawarkan akses yang luas dari informasi dan komunikasi serta hiburan dengan penggunaan yang mudah dan hemat biaya. Pengguna juga ingin selalu tersambung dimanapun dan kapanpun.

Pada umumnya, operator mencari cara yang paling mudah dan fleksibel untuk merespon kesempatan bisnis yang ada. Operator juga ingin memberikan pengalaman user yang tak henti dan konsisten dimanapun dan bagaimanapun layanan dapat di akses. Operator selalu memperhitungkan biaya dan cara mewujudkan bisnis model mereka yang lebih efisien.

*IP Multimedia Subsystem* (IMS) adalah kerangka arsitektur untuk layanan multimedia berbasis Internet Protocol (IP) yang didesain dan dikeluarkan oleh *3<sup>rd</sup> Generation Partnership Project* (3GPP). IMS menggunakan protokol yang dikembangkan oleh *Internet Engineering Task Force* (IETF) yaitu badan standarisasi yang mengembangkan standar Internet. Karena IMS dibangun diatas protokol IETF, IMS mempunyai sistem multimedia yang lengkap dan kuat. [4]. Bagi pengguna, layanan berbasis IMS menjawab semua keinginan umum para pengguna. IMS memungkinkan komunikasi pengguna ke pengguna, dan pengguna ke layanan dalam model yang bervariasi, termasuk suara, teks, gambar, video, atau kombinasinya, secara personal dalam jalur yang terkendali.

Bagi operator, IMS menggunakan konsep lapisan arsitektur yang selangkah lebih jauh, dengan menggunakan konsep arsitektur horizontal, dimana memungkinkan layanan dan fungsi umum digunakan kembali untuk banyak aplikasi. Arsitektur horizontal dalam IMS juga memungkinkan *interoperability* dan *roaming*, serta menyediakan *control*, *charging* dan

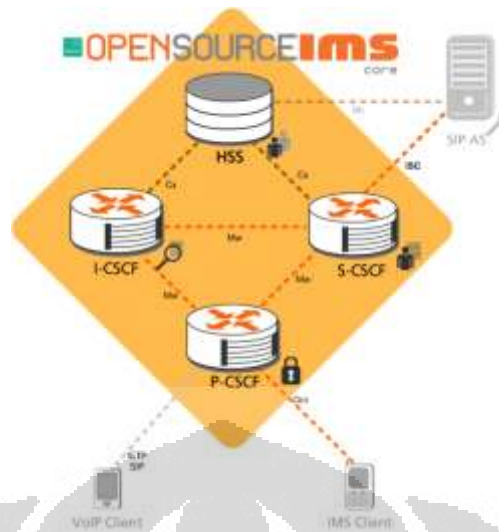
keamanan. IMS juga terintegrasi dengan baik dengan jaringan suara dan data yang ada, yang mengadopsi dengan baik manfaat kunci dari domain IT. [5]. Ini membuat IMS adalah kunci yang memungkinkan terjadinya *Fixed Mobile Convergence* (FMC). Karena alasan-alasan ini, IMS menjadi solusi yang dipilih untuk operator bisnis multimedia baik tetap (*fixed*) dan bergerak (*mobile*).



Gambar 2.1 Konsep Integrasi Horizontal Layanan [5]

### 2.1.1. Komponen-komponen IMS

Komponen-komponen IMS adalah komponen yang memiliki fungsi dalam arsitektur IMS. Komponen-komponen utama dari IMS adalah *Proxy Call Session Control Function* (P-CSCF), *Interrogating Call Session Control Function* (I-CSCF), *Service Call Session Control Function* (S-CSCF) dan *Home Subscriber Server* (HSS).



Gambar 2.2 Jaringan IMS [6]

#### 2.1.1.1. Proxy Call Session Control Function (P-CSCF)

P-CSCF adalah titik pertama dari jalur pensinyalan antara terminal/perangkat IMS dengan Jaringan IMS. P-CSCF beraksi sebagai pintu masuk dan keluar server SIP. Ini berarti, semua permintaan yang dilakukan oleh IMS terminal atau kebalikannya dari P-CSCF ke IMS terminal. P-CSCF meneruskan permintaan SIP dan memberikan respon kearah yang tujuan.

P-CSCF mempunyai beberapa fungsi, diantaranya berhubungan dengan keamanan. P-CSCF menghasilkan nomor IPsec yang berhubungan dengan keamanan kearah terminal IMS. IP sec ini memberikan proteksi yang terintegrasi. Saat P-CSCF memeriksa pengguna (sebagai bagian dari pengadaan keamanan) , P-CSCF meminta identitas pengguna untuk digunakan di titik lain di jaringan. Titik lain di jaringan tidak melakukan autentifikasi lagi, karena sudah dilakukan oleh P-CSCF. Selain itu, P-CSCF melakukan pengecekan kebenaran permintaan SIP yang dikirimkan oleh terminal. Pengecekan ini menjaga agar terminal IMS terjaga dari permintaan SIP yang tidak sesuai dengan kaidah SIP.

Di dalam P-CSCF juga terdapat *compressor* dan *decompressor* pesan SIP. Ukuran pesan SIP dapat menjadi besar, karena SIP adalah protokol berbasis teks. Ketika pesan SIP dapat ditransmisikan melalui koneksi pita lebar dalam waktu singkat, transmisi pesan SIP yang besar

melalui kanal yang kecil akan menggunakan waktu beberapa saat. Terdapat mekanisme yang digunakan untuk mengurangi waktu untuk mentransmisikan pesan SIP. Pesan SIP dimampatkan di satu sisi, dikirimkan, lalu dikembalikan kembali pada satu sisi yang lain.

P-CSCF dapat memasukan PDF (*Policy Decision Function*). PDF dapat diintegrasikan dengan P-CSCF atau diimplementasikan sebagai unit tersendiri. PDF memeriksa sumber jalur media, dan mengatur *Quality of Service* (QoS) melalui jalur media. P-CSCF juga menghasilkan informasi mengenai *charging* ke arah pengumpul informasi *charging*.

Setiap P-CSCF melayani sejumlah terminal IMS, tergantung pada kapasitas dari suatu titik jaringan.

#### **2.1.1.2. Interrogating Call Session Control Function (I-CSCF)**

I-CSCF adalah SIP *proxy* yang terletak di tepi domain administrasi. Alamat dari I-CSCF terdapat di DNS (*Domain Name System*). Ketika SIP server mengikuti prosedur SIP untuk mencari tempat SIP selanjutnya untuk suatu pesan, SIP server mengambil alamat dari I-CSCF sebagai domain tujuan.

Disamping berfungsi sebagai server *proxy* SIP, I-CSCF mempunyai hubungan dengan SLF (*Subscriber Location Function*) dan HSS. Hubungan ini didasarkan oleh protokol Diameter. I-CSCF mengambil informasi lokasi pengguna dan mencari jalur permintaan SIP menuju tujuan yang diinginkan (S-CSCF).

Selain itu, I-CSCF dapat melakukan enkripsi dari pesan SIP yang mengandung informasi penting mengenai domain, seperti jumlah server dalam domain, nama DNS dan kapasitasnya. Fungsi ini didasarkan oleh THIG (*Topology Hiding Internet Gateway*). THIG secara fungsi adalah pilihan dan tidak diterapkan oleh kebanyakan jaringan.

#### **2.1.1.3. Serving Call Session Control Function (S-CSCF)**

S-CSCF adalah titik sentral dari jalur pensinyalan. S-CSCF sesungguhnya adalah server SIP, tetapi juga berperan sebagai pengendali sesi. Selain itu, S-CSCF juga berperan sebagai pendaftar SIP. Ini berarti S-

CSCF mengetahui hubungan antara lokasi pengguna, dan catatan alamat pengguna SIP (*Public User Identity*).

S-CSCF juga menggunakan Diameter untuk berhubungan dengan HSS. Alasan utama S-CSCF berhubungan dengan HSS adalah :

- a. Untuk mengunduh dari HSS vektor autentifikasi dari pengguna yang mencoba mengakses IMS. S-CSCF menggunakan vektor ini untuk mengautentifikasi pengguna.
- b. Untuk mengunduh profil pengguna dari HSS. Profil pengguna ini memasukan profil layanan yang berupa pemicu pesan SIP agar di rutekan melalui satu server aplikasi atau lebih.
- c. Untuk menginformasikan HSS bahwa S-CSCF tersebut yang dialokasikan pengguna selama masa registrasi.

Semua pensinyalan SIP terminal IMS yang sedang mengirim dan menerima, melintasi S-CSCF yang dialokasikan. S-CSCF memeriksa setiap pesan SIP dan menentukan apakah pensinyalan SIP seharusnya mengunjungi satu server aplikasi atau lebih dan merutekan menuju tujuan terakhir. Server aplikasi diharapkan menyediakan layanan kepada pengguna.

Salah satu dari fungsi utama dari S-CSCF adalah menyediakan layanan pengurutan SIP. Jika pengguna menekan nomor telepon, bukan SIP URI (*Uniform Resource Identifier*), S-CSCF menyediakan layanan penerjemahan, berdasarkan DNS E.164 *Translation Number*.

S-CSCF juga menjalankan peraturan dari operator jaringan. Sebagai contoh, pengguna mungkin tidak diautorisasi untuk membangun beberapa jenis sesi. S-CSCF menjaga pengguna dari operasi yang tidak diijinkan.

#### **2.1.1.4. Home Subscriber Server (HSS)**

*Home Subscriber Server* (HSS) adalah tempat penyimpanan utama untuk informasi yang berkaitan dengan pengguna. HSS mengandung semua data yang berkaitan dengan pengguna yang dibutuhkan untuk mengadakan sesi multimedia. Data ini termasuk informasi lokasi,

informasi keamanan (termasuk informasi autentifikasi dan otorisasi), informasi profil pengguna (termasuk layanan dimana user telah terdaftar) dan informasi mengenai S-CSCF yang telah dialokasikan untuk pengguna.

Jaringan mungkin mempunyai lebih dari satu HSS, ketika jumlah pendaftar terlalu besar untuk ditangani oleh satu HSS.

## 2.2. Protokol-protokol yang Digunakan Dalam IMS

### 2.2.1. SIP Protocol

SIP adalah protokol yang dikeluarkan oleh IETF untuk membangun dan mengatur sesi multimedia melewati jaringan IP. SIP mengikuti model *client-server* yang terkenal, begitu banyak digunakan oleh banyak protokol yang dikembangkan oleh IETF. Pembuat SIP meminjam prinsip desain dari SMTP (*Simple Mail Transfer Protocol*) dan khususnya HTTP (*Hypertext Transfer Protocol*). SIP mewarisi banyak karakteristik dari dua protokol ini. Ini adalah kekuatan dari SIP, karena HTTP dan SMTP adalah protokol yang paling sukses di Internet. SIP telah terpilih sebagai protokol pengontrol sesi untuk IMS. SIP telah terbukti membuat kemudahan dalam membangun layanan baru yang dibawa dengan bobot yang tidak besar. Karena SIP didasarkan oleh HTTP, para pengembang layanan berbasis SIP dapat menggunakan kerangka layanan yang mereka kembangkan untuk HTTP, seperti CGI (*Common Gateway Interface*) dan Java.

SIP mendukung lima fungsi dalam membangun dan mengakhiri komunikasi multimedia, yaitu:

- a. *User Location*  
Penentuan dari tujuan akhir sistem yang akan digunakan dalam komunikasi
- b. *User Availability*  
Penentuan keinginan dari pihak yang dipanggil untuk terlibat & masuk dalam komunikasi.
- c. *User Capabilities*  
Penentuan media dan parameter media yang akan digunakan
- d. *Session Setup*

Pembuatan panggilan dan negosiasi pembuatan parameter sesi baik disisi pemanggil dan yang dipanggil.

e. *Session Management*

Parameter dari sesi dimodifikasi, data dikirimkan, layanan dicabut, dan sesi dihentikan.

### 2.2.1.1. Arsitektur SIP

Ada dua komponen yang paling fundamental terdapat didalam SIP, yaitu user agent dan SIP server.

a. *User Agent*

*User Agent* adalah dua devais pengguna, yang melakukan panggilan dan tujuan panggilan tersebut. Ada dua komponen dari *user agent*, yaitu *client* dan *server*. Ketika *user agent* melakukan permintaan panggilan (menginisiasi sesi), ini disebut *User Agent Client (UAC)*, dan *user agent* yang memberikan respon adalah *User Agent Server (UAS)*. Karena *user agent* akan mengirim pesan dan memberikan balasan, status ini dibalik dan ditukar.

b. *SIP Server*

Server SIP digunakan untuk memecahkan *username* menjadi alamat IP, maka permintaan yang dikirim dari *user agent* ke yang lain dapat langsung dikirim dengan baik. *user agent* mendaftar dengan server SIP, menyediakan *user agent* dengan nama dan alamat IP, dengan demikian menetapkan lokasi terbaru dalam jaringan. Server SIP juga memeriksa apakah *user agent online* atau tidak, sehingga *user agent* lain dapat melihat apakah mereka ada dan dapat diundang untuk masuk ke sesi. Karena *user agent* mungkin tidak mengetahui alamat IP dari *User Agent* lain, permintaan dibuat kepada SIP server untuk mengundang *user* lain ke dalam sesi. Server SIP lalu mengidentifikasi apakah orang tersebut *online* atau tidak. Jika *online*, maka membandingkan *username* dengan alamat IP milik mereka untuk menentukan lokasinya. Jika pengguna bukan bagian dari domain tersebut, maka digunakan server SIP yang berbeda. Maka permintaan tadi akan diteruskan ke server lainnya.

SIP server mempunyai beberapa peran, yaitu *Registrar Server*, *Proxy Server*, dan *Redirect Server*.

- *Registrar Server*

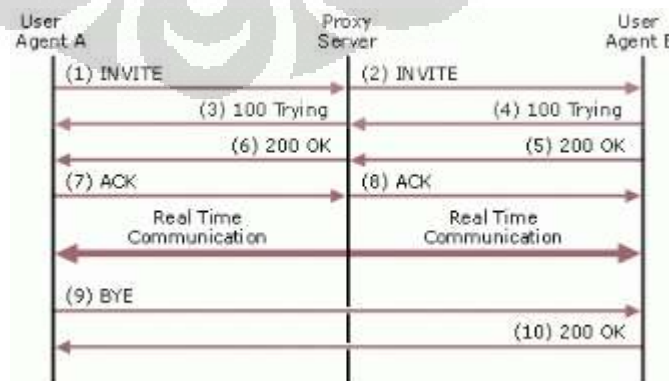
Digunakan untuk mendaftarkan lokasi dari *user agent* yang telah masuk dalam jaringan. *Registrar Server* memperoleh alamat IP dari pengguna dan menghubungkannya dengan *username* dalam sistem. Ini membuat sebuah direktori dari semua yang telah masuk dalam jaringan dan dimana lokasi mereka. Ketika seseorang menginginkan membuat sesi dengan salah satu dari pengguna, informasi dari *Registrar Server* ditunjuk, dengan demikian alamat IP yang diinginkan diperoleh untuk sesi tersebut. [7]



Gambar 2.3 SIP server sebagai *Registrar Server* [7]

- *Proxy Server*

*Proxy server* digunakan untuk meneruskan permintaan ke komputer lain / server lain. Jika server SIP menerima permintaan dari *client*, *Proxy server* dapat meneruskan permintaan ke server SIP lain didalam jaringan. Ketika berfungsi sebagai *Proxy server*, server SIP dapat menyediakan fungsi seperti kendali akses jaringan, keamanan, autentifikasi dan otorisasi. [7]

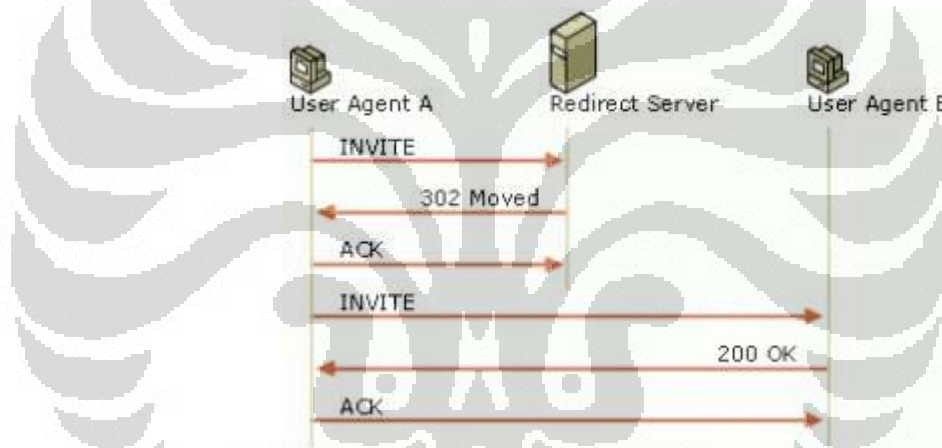


Gambar 2.4 SIP server sebagai *Proxy Server* [7]



- *Redirect Server*

Digunakan oleh server SIP untuk meneruskan kepada *user agent* yang ingin dihubungi. Jika *user agent* membuat permintaan, *Redirect server* dapat memberikan respon dengan alamat IP dari *user agent* yang sedang dihubungi. Ini berbeda dengan *proxy server*, yang meneruskan permintaan dalam kepentingan pemanggil, *Redirect server* memberitahu *user agent* untuk menghubungi pemanggil. *Redirect server* juga mempunyai kemampuan mencabangkan panggilan, dengan memisahkan panggilan ke beberapa lokasi berbeda. Jika panggilan dibuat untuk suatu pengguna, maka dapat dipisahkan menjadi beberapa nomor dilokasi berbeda. [7]



Gambar 2.5 SIP server sebagai *Redirect Server* [7]

### 2.2.1.2. SIP Message

SIP message terdiri dari 3 jenis , yaitu *request*, *response* dan *acknowledgement*.

#### a. Request

- REGISTER

Digunakan ketika *user agent* pertama *online* dan mendaftarkan alamat SIP dan alamat IP ke server Register

- INVITE

Digunakan untuk mengundang *user agent* lain untuk berkomunikasi, lalu membangun sesi SIP antara mereka.

- ACK

Digunakan untuk menerima sesi dan mengkonfirmasi pertukaran pesan yang dibutuhkan

- OPTION

Digunakan untuk mengambil informasi dari kemampuan *user agent* lain, sehingga sesi tersebut dapat dibangun diantara mereka. Ketika informasi ini disediakan, sesi tidak secara otomatis dibangun sebagai hasilnya.

- SUBSCRIBE

Digunakan untuk meminta *update* informasi kehadiran *user agent* lain. Ini digunakan untuk memperoleh informasi terbaru seperti, apakah *user agent online, busy* atau *offline*.

- NOTIFY

Digunakan untuk mengirim informasi terbaru dari status terkini *user agent*. Informasi kehadiran ini memberitahukan apakah *user agent* mempunyai status *online*, *busy*, atau *offline*.

- CANCEL

Digunakan untuk membatalkan permintaan yang tertunda tanpa menghentikan sesi.

- BYE

Digunakan untuk mengakhiri sesi. Baik *user agent* yang memulai atau dipanggil, dapat menggunakan perintah ini setiap waktu.

b. *Response*

Ketika *request* dilakukan pada SIP server atau *user agent* lain, ada beberapa kemungkinan respon yang dapat dikirim balik. Respon-respon ini dikelompokkan ke dalam 6 katagori yang berbeda, dengan tiga buah kode angka respon, dimulai dengan satu angka yang mewakili kategori tersebut. Kategori tersebut dan code responnya

adalah *Informational (1xx)*, *Success (2xx)*, *Redirection (3xx)*, *Client Error (4xx)*, *Server Error (5xx)* dan *Global Failure (6xx)*.

- **Informational (1xx)**

Kode ini menunjukkan bahwa *request* telah sampai dan sedang diproses.

- **Success (2xx)**

Menunjukkan bahwa *request* telah dijawab dan diterima.

- **Redirection (3xx)**

Menunjukkan bahwa *request* tidak dapat dilengkapi dan membutuhkan langkah tambahan.

- **Client Error (4xx)**

Menunjukkan bahwa *request* mengandung kesalahan, maka server tidak dapat memproses *request*.

- **Server Error (5xx)**

*Request* telah diterima, tetapi server tidak dapat memproses. Kesalahan ini terjadi karena masalah di server, dan tidak menyatakan bahwa server lain tidak dapat memproses *request*.

- **Global Failure (6xx)**

Kode ini menyatakan bahwa *request* telah diterima dan server tidak dapat memprosesnya. Kesalahan seperti ini disebabkan karena masalah terjadi di setiap server, sehingga *request* tak akan diteruskan pada server lain untuk diproses. [7]

### 2.2.1.3. SIP dan SIPS Uniform Resource Indicators

SIP atau SIPS URI adalah identitas dari pengguna komunikasi. SIP dan SIPS URI dapat ditempatkan di halaman web, isi pesan *email*, atau literatur yang tertulis. SIP dan SIP URI mengandung informasi yang dibutuhkan untuk memulai suatu komunikasi dengan pengguna komunikasi.

Penamaan SIPS pada SIPS URI menunjukkan bahwa sumber harus dihubungi dengan metode yang lebih aman (*SIP Secure*). Komunikasi yang lebih aman digunakan untuk menghubungi pengguna, dimana

mekanisme keamanan yang lebih spesifik diterapkan pada suatu domain. Setiap pengguna SIP URI dapat ditingkatkan menjadi SIPS URI dengan merubah skema, dan hanya jika menginginkan komunikasi yang lebih aman.

Format yang digunakan dalam SIP mirip dengan alamat email.

**[sip:user:password@host:port](#)**

- user** : adalah identitas dari pengguna pada host yang menyediakan alamat. Penulisan user bersifat *case sensitive*
- Password** : adalah kata kunci yang digunakan oleh user. Penulisan password tidak direkomendasikan karena password merupakan bagian dari keamanan. Jika tidak dituliskan password, maka tidak akan terjadi masalah apapun. Penulisan password bersifat *case sensitive*
- Host** : adalah penyedia dari alamat SIP. Penulisan host adalah diwajibkan.
- Port** : adalah nomor port dimana *Request* akan dikirimkan.

### 2.2.2. SDP

Pesan SIP yang digunakan untuk membuat sesi membawa deskripsi dari sesi yang memungkinkan peserta komunikasi menyetujui tipe-tipe media yang digunakan. Deskripsi sesi ini diatur dengan menggunakan format SDP.

Tujuan dari SDP adalah menyampaikan informasi mengenai aliran media dalam sesi multimedia agar penerima menyetujui untuk berpartisipasi dalam sesi yang ingin dibangun. SDP pada dasarnya digunakan untuk kerja antar jaringan, meskipun secara umum cukup dapat mendeskripsikan konferensi pada lingkungan jaringan lain. Deskripsi sesi yang dibawa oleh SDP meliputi:

- Nama sesi dan tujuannya
- Waktu aktif dari sesi.
- Media yang ada didalam sesi
- Informasi yang dibutuhkan untuk menerima media tersebut (alamat, *port*, format, dan lainnya)

Karena sumber daya yang dibutuhkan untuk berpartisipasi dalam sesi mungkin terbatas, maka beberapa informasi tambahan juga dibutuhkan, seperti informasi mengenai *bandwidth* yang dibutuhkan dalam sesi, dan informasi kontak orang yang bertanggung jawab dalam sesi.

Deskripsi sesi yang dijelaskan oleh SDP berisi baris-baris teks yang mempunyai format: `<type>=<value>`. `<Type>` harus satu karakter (case sensitive) dan `<value>` adalah teks yang terstruktur tergantung dari `<type>`.

Berikut ini adalah contoh-contoh SDP yang ada [8]

#### ***Session description***

*v= (protocol version)*

*o= (originator and session identifier)*

*s= (session name)*

*i=\* (session information)*

*u=\* (URI of description)*

*e=\* (email address)*

*p=\* (phone number)*

*c=\* (connection information -- not required if included in all media)*

*b=\* (zero or more bandwidth information lines)*

*One or more time descriptions ("t=" and "r=" lines; see below)*

*z=\* (time zone adjustments)*

*k=\* (encryption key)*

*a=\* (zero or more session attribute lines)*

*Zero or more media descriptions*

#### ***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)*

*c=\* (connection information -- optional if included at session level)*

*b=\* (zero or more bandwidth information lines)*

*k=\* (encryption key)*

*a=\* (zero or more media attribute lines) [8]*

### 2.2.3. Diameter

Diameter adalah protokol yang menyediakan kerangka Autentifikasi, Autorisasi dan Akuntansi untuk banyak aplikasi seperti akses jaringan dan mobilitas IP. [9]. Diameter dispesifikasikan sebagai protokol dasar dan pengaturan untuk aplikasi Diameter yang melengkapi fungsi protokol dasar. Protokol dasar mengandung fungsi yang dasar dan diimplementasikan di semua titik Diameter, tidak tergantung pada aplikasi yang spesifik. Aplikasi-aplikasi adalah perluasan dari fungsi-fungsi dasar yang disesuaikan untuk kegunaan khusus dari Diameter di lingkungan yang spesifik.

Diameter berjalan diatas transport yang dapat diandalkan yang menawarkan kendali kemacetan. Di sisi lain, Diameter tidak berjalan diatas UDP (*User Datagram Protocol*). Tidak seperti RADIUS, pesan Diameter yang hilang dikirimkan kembali pada setiap titik. Diameter menyediakan pesan yang berganti di level aplikasi, yang mengawasi status koneksi dan mengijinkan pemulihan dari kesalahan kondisi. Diameter juga mengijinkan penghitungan pesan agar dirutekan ke server yang berbeda.

Diameter mendefinisikan beberapa fungsi dalam mewujudkan fungsi AAA, yaitu :

- *Diameter Client*  
Suatu fungsi yang terletak di tepi jaringan, yang melakukan kendali akses.
- *Diameter server*  
Suatu fungsi yang menangani Autentifikasi, Autorisasi dan Akuntansi dari *Request* untuk pengalaman ulang.
- *Proxy*  
Adalah suatu fungsi yang menjadi pemilih pengaturan yang berhubungan dengan penggunaan sumber daya dan perlengkapan. Proxy dapat merubah pesan untuk mengimplementasikan pemilihan

peraturan seperti pengendalian penggunaan sumber daya, menyediakan kendali hak masuk, dan perlengkapan.

- *Relay*

Suatu fungsi yang meneruskan pesan Diameter, didasarkan pada informasi yang berhubungan dengan rute, dan pengalamatan ulang isi tabel penjaluran. *Relay* bersifat transparan. *Relay* dapat mengubah pesan Diameter hanya dengan memasukan atau menghapus data yang berhubungan dengan penjaluran.

- *Redirect Agent*

Suatu fungsi yang menghubungkan *client* ke server dan mengizinkan mereka untuk berkomunikasi secara langsung.

- *Translation Agent*

Suatu fungsi yang menyelenggarakan penerjemahan protokol antara Diameter dan Protokol AAA lainnya, seperti Radius.

- *Diameter Node*

Suatu fungsi yang mengimplementasikan protocol Diameter dan bertindak baik sebagai *client* Diameter dan server Diameter, *Relay*, *redirect agent*, atau *translation agent*.

### 2.3. TEL URI

Format lain yang digunakan dalam berkomunikasi dalam IMS adalah format TEL URI, sesuai dengan standar RFC 3966. Berdasarkan RFC 3966, TEL URI adalah salah satu skema URI yang menjelaskan identitas dari nomor telepon. Nomor telepon adalah rangkaian angka desimal yang menjelaskan titik tujuan dari jaringan. Nomor telepon berisi informasi yang dibutuhkan untuk menyalurkan sebuah panggilan ketempat tujuan. Titik tujuan dari nomor yang dijelaskan oleh TEL URI tidak terbatas. Titik tujuan tersebut bisa jaringan telepon umum, jaringan telepon pribadi, ataupun internet. TEL URI juga tidak menunjuk pada sebuah devais fisik secara spesifik, hanya menunjukan sebuah nomor telepon.

Secara umum, nomor telepon memiliki dua makna konsep, yaitu sebagai pengenal, dan sebagai rangkaian angka telepon (*dial string*).

a. Nomor telepon sebagai Pengenal

Nomor telepon di kenal sebagai alamat atau pengenal dari tujuan sebuah jaringan yang spesifik. Dalam jaringan telepon umum, penomoran ini mengikuti peraturan E.164 sedangkan untuk nomor pribadi mengikuti peraturan pemilik rencana penomoran. Pelanggan mengenalkan nomor ini, sehingga mereka dapat dipanggil.

b. Nomor telepon sebagai rangkaian angka telepon (*dial string*).

Rangkaian angka telepon adalah semua nomor, simbol yang ditekan oleh pengguna saat ingin melakukan panggilan. Rangkaian ini dikenal sebagai konfigurasi dari jaringan. Ini digunakan untuk menghasilkan sebuah alamat tujuan, sehingga panggilan dapat sampai ke penerima. [10]

Format TEL URI dapat merepresentasikan nomor telepon dalam format global, yaitu dengan diawali dengan karakter “+”. Format global harus di ikuti dengan Kode Negara dan Kode nasional. Sementara format lokal adalah format yang digunakan untuk sekelompok kecil penggunaan, seperti sebuah area geografis, atau *Private Branch Exchange* (PBX). Format lokal harus memiliki parameter “*phone-context*” yang mengidentifikasi lingkup penggunaan nomor telepon tersebut. [10]

Format Global:

<tel:+1-212-555-0293>

Format Lokal :

<tel:7042;phone-context=example.com> [10]

Format TEL URI dibutuhkan untuk melakukan panggilan dari terminal IMS ke telepon PSTN, karena nomor PSTN hanya direpresentasikan dengan nomor angka. Di lain sisi, TEL URI juga dibutuhkan ketika pelanggan PSTN ingin melakukan panggilan kepada pelanggan IMS, karena pelanggan PSTN hanya dapat menekan nomor angka.



#### 2.4. E.164 Number Mapping (ENUM)

Banyak jaringan komunikasi dibangun untuk satu bentuk komunikasi, dan tidak dapat digunakan oleh bentuk komunikasi yang lainnya. Walaupun internet juga termasuk jaringan yang khusus dalam menyediakan komunikasi digital, internet lebih fleksibel dalam kemampuannya mengkodekan berbagai macam format komunikasi, sehingga mendukung interaksi format komunikasi lainnya melalui internet. Dalam hal ini, banyak jaringan komunikasi dipetakan menuju aplikasi internet, sehingga dapat didistribusikan sebagai aplikasi diatas Internet. Dan untuk pertumbuhan layanan *Voice over IP* (VoIP), dapat mentransmisikan suara melalui internet saja belum cukup. Dibutuhkan integrasi lebih baik antara jaringan telepon konvensional dengan VoIP.

Bentuk komunikasi konvensional yang menggunakan konsep sistem switching telah berkembang menjadi sistem penomoran global, dimana setiap alat yang tersambung dengan jaringan telepon mempunyai alamat nomor yang unik. Penomoran ini diatur oleh *International Telecommunication Union* (ITU), dan rekomendasi E.164 yang membagi awalan penomoran kepada setiap kode administrasi suatu Negara.

Untuk membentuk integrasi penggunaan layanan suara dalam dunia Internet dengan jaringan telepon, dukungan kepada penomoran E164 adalah hal yang penting. Untuk membuat telepon via internet benar-benar berguna, maka dunia telepon internet harus dapat berkomunikasi dengan jaringan telepon konvensional. Telepon internet harus dapat menerima dan melakukan panggilan ke devais telepon apapun, apakah devais tersebut tersambung ke Internet, tersambung ke jaringan telepon konvensional atau tersambung dengan jaringan lain yang berhubungan dengan jaringan telepon konvensional. Untuk itu, salah satu kondisi yang harus dilengkapi adalah bahwa setiap devais Internet yang mendukung operasi internet harus mempunyai bentuk alamat nomor telepon. [11]

Telephone Number Mapping (ENUM) adalah proses untuk mempersatukan sistem nomor telepon dari jaringan telepon konvensional dengan pengalamatan di Internet dan identifikasi nama. Nomor telepon secara sistematis diatur di dalam E.164, sementara Internet menggunakan *Domain*

*Name System* (DNS) untuk menyambungkan antara nama domain antara alamat IP dengan sumber daya informasi lainnya. Sistem penomoran dengan menggunakan ENUM menyediakan fasilitas untuk menentukan server komunikasi internet yang mampu untuk melayani pemberian nomor telepon dengan melakukan pengecekan di dalam *Domain Name System*(DNS).

ENUM menyediakan mekanisme yang hemat biaya, yang memungkinkan sebuah IP mempunyai banyak layanan. Dengan ENUM operator dapat saling berhubungan satu sama lain dengan sederhana baik di bentuk bisnis dan operasional.



Gambar 2.6 . Komponen CSCF Akses Database ENUM Server [12]

#### 2.4.1. Aplikasi ENUM

ENUM digunakan dalam beberapa aplikasi, antara lain: [13]

1. Sambungan telepon jaringan VoIP secara penuh.
2. ENUM digunakan saat pengguna VoIP menggunakan format nomor E.164.

Selama proses yang terjadi antar server SIP, database ENUM diakses untuk mendapatkan alamat berformat SIP tujuan.

3. Sambungan panggilan dari PSTN ke VoIP melalui jaringan suara terpadu.

ENUM digunakan saat terjadi panggilan dari pelanggan PSTN ke pelanggan dengan telepon SIP. PSTN akan menghubungi *Media Gateway* mengirimkan pesan IAM dalam format ISUP. *Media Gateway* akan meneruskan pesan SIP INVITE kepada SIP Proxy. SIP Proxy akan meminta *query* dari ENUM database. *Query* ini akan digunakan untuk menghubungi pelanggan telepon SIP.

4. Dasar komunikasi terpadu untuk jenis komunikasi yang berbeda, seperti Suara, Internet Fax, dan Email.
5. Translasi nomor dalam IMS
6. Akses website menggunakan nomor E.164

Enum mempunyai beberapa jenis layanan yang didukung, yaitu:

Tabel 2.1 Jenis-Jenis Layanan yang di dukung ENUM Server

Service Name	Service Field	URI
SIP	E2U+sip	<a href="mailto:Sip:faisal@tm.com.my">Sip:faisal@tm.com.my</a>
H.323	E2U+h323	<a href="mailto:h323:faisal@tm.com.my">h323:faisal@tm.com.my</a>
Email	E2u+email	<a href="mailto:faisal@tm.com.my">mailto:faisal@tm.com.my</a>
Internet Fax	E2U+ifax	<a href="mailto:faxfaisal@tm.com.my">mailto:faxfaisal@tm.com.my</a>
PSTN	E2U+tel	<a href="tel:+8741664">Tel:+8741664</a>
Web Page	E2U+web	<a href="http://tm.com.my">http://tm.com.my</a>

#### 2.4.2. Konversi format E.164 ke ENUM

Dalam melakukan konversi format E.164 ke dalam format yang dapat diakses dalam ENUM, terdapat sebuah aturan yang memudahkan konversinya, seperti di bawah ini:

1. Buang semua karakter kecuali nomor. Sebagai contoh, nomor yang ingin digunakan adalah +874-1664. Dengan langkah ini, hasil akhirnya adalah 8741664.
2. Letakan titik diantara setiap angka. Contoh 8.7.4.1.6.6.4
3. Balik urutan dari angka tersebut. Contoh 4.6.6.1.4.7.8
4. Tambahkan string “.e164.arpa” pada akhir urutan. Contoh: 4.6.6.1.4.7.8.e164.arpa. Nama ini digunakan untuk panggilan daftar Naming Authority Pointer (NAPTR) Resource [13]

#### 2.4.3. ENUM NAPTR DATA

NAPTR (*Naming Authority Pointer*) adalah sebuah jenis catatan data yang digunakan dalam DNS. NAPTR menjelaskan berbagai macam informasi dalam data yang dimiliki di dalam DNS. Informasi-informasi tersebut adalah

Tipe (*type*), Urutan (*Order*), Preferensi (*Preferences*), Tanda (*Flag*), Layanan (*Services*), RegExp, dan Pengganti (*Replacement*). Untuk lebih jelasnya, berikut ini adalah contoh penulisan data NAPTR:

```
$ORIGIN 3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa.
```

```
NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:info@example.com!".
```

```
NAPTR 10 101 "u" "E2U+h323" "!^.*$!h323:info@example.com!".
```

```
NAPTR 10 102 "u" "E2U+msg" "!^.*$!mailto:info@example.com!"
```

Angka *3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa* adalah data nomor telepon, yang akan dicocokkan oleh hasil konversi dari format tel ke format yang digunakan di dalam ENUM. Domain dibelakang angka tersebut adalah domain yang digunakan untuk menunjukkan penggunaan ENUM didalam data tersebut.

Dibawah data tersebut, terdapat parameter-parameter dalam format NAPTR. NAPTR harus tertulis agar menunjukkan jenis data di dalam DNS. Angka pertama adalah menunjukkan urutan, yang mana dari data NAPTR yang harus diproses terlebih dahulu. Angka yang rendah harus diproses terlebih dahulu. Ketika sebuah data cocok dengan aturan ini, maka *client* harus tidak memperhatikan parameter lain dalam data NAPTR, kecuali Tanda (*flags*).

Angka kedua adalah preferensi, dimana memberikan tanda, data mana yang harus diproses terlebih dahulu, ketika terdapat urutan (*order*) yang sama. Angka ini berperan seperti bobot, agar dapat memilih, jenis layanan mana yang seharusnya dipilih terlebih dahulu.

Parameter selanjutnya adalah sebuah karakter. Karakter ini mengendalikan aspek penulisan ulang dan interpretasi dari data. Karakter ini adalah sebuah karakter dari A-Z. Sampai saat ini hanya 4 karakter yang telah didefinisikan menjadi tanda (*flag*) NAPTR. Huruf "U" untuk menandakan bahwa keluaran NAPTR adalah URI. Sementara huruf "S" menandakan bahwa langkah selanjutnya adalah mencari data SRV. Huruf A menandakan bahwa pencarian selanjutnya seharusnya adalah alamat (*Address*), baik A, AAAA, atau data A6. Huruf P menandakan bahwa bahwa sisa dari algoritma di sisi aplikasi harus dibawa keluar dengan standar protocol yang spesifik.

Parameter selanjutnya adalah layanan. Layanan yang spesifik dapat ditulis di dalam penulisan data NAPTR. Ini juga dapat menentukan protokol yang digunakan dalam layanan tersebut. Contohnya adalah E2U+SIP yang berarti menggunakan protokol SIP. Parameter lainnya adalah Regexp. Regexp adalah bentuk dari pengganti yang diterapkan kepada bentuk awal. Sementara Pengganti (*Replacement*) adalah *output* pengganti dari permintaan data NAPTR. [14]

## 2.5. Lama Waktu dalam Respon Pencarian

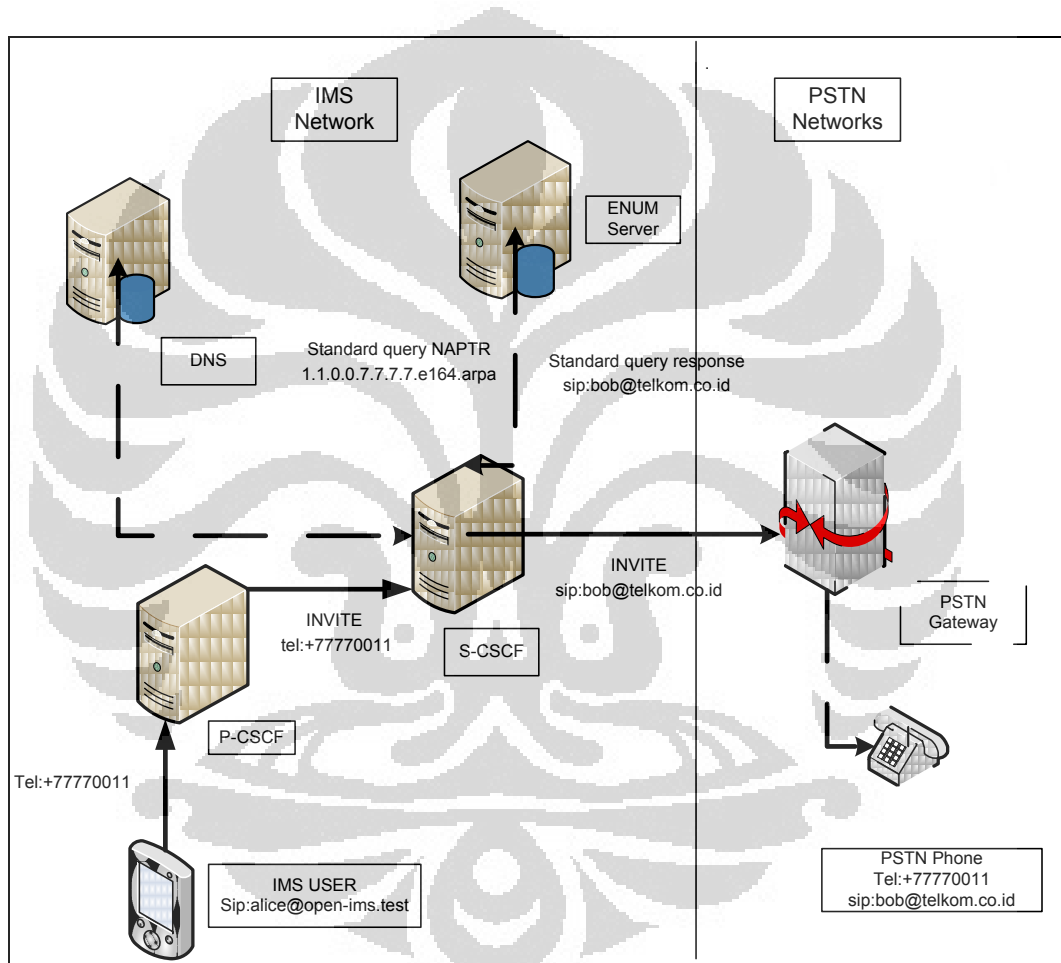
Latensi yang disebabkan oleh proses ENUM adalah salah satu komponen dari latensi total pembangunan panggilan. Untuk menyediakan VoIP dengan kualitas yang dapat dibandingkan dengan layanan PSTN, pensinyalan total panggilan VoIP harus tidak lebih besar dari pensinyalan pembentukan panggilan dalam sistem SS7 (*Signaling System 7*). Proses pencarian data ENUM dapat dibandingkan dengan proses perutean dalam sistem PSTN. Total waktu respon untuk sebuah pesan memiliki rata 205-218 ms, dan 337-355-ms dalam persentase 95 persen [3]. Total waktu respon ini adalah batasan maksimum dalam pembentukan suatu panggilan. Hal ini berarti waktu pencarian dengan ENUM, termasuk lama proses server ENUM dan waktu perutean perjalanan seharusnya hanya memiliki waktu yang sangat pendek.

## BAB 3

### SKENARIO PENGGUNAAN ENUM SERVER PADA OPEN IMS CORE

#### 3.1. Skenario Penggunaan ENUM Server dalam Jaringan IMS

Pada umumnya, ENUM Server digunakan ketika akan terjadinya koneksi atau panggilan dari jaringan yang berbeda, yaitu jaringan PSTN ke jaringan berbasis Internet, atau sebaliknya.



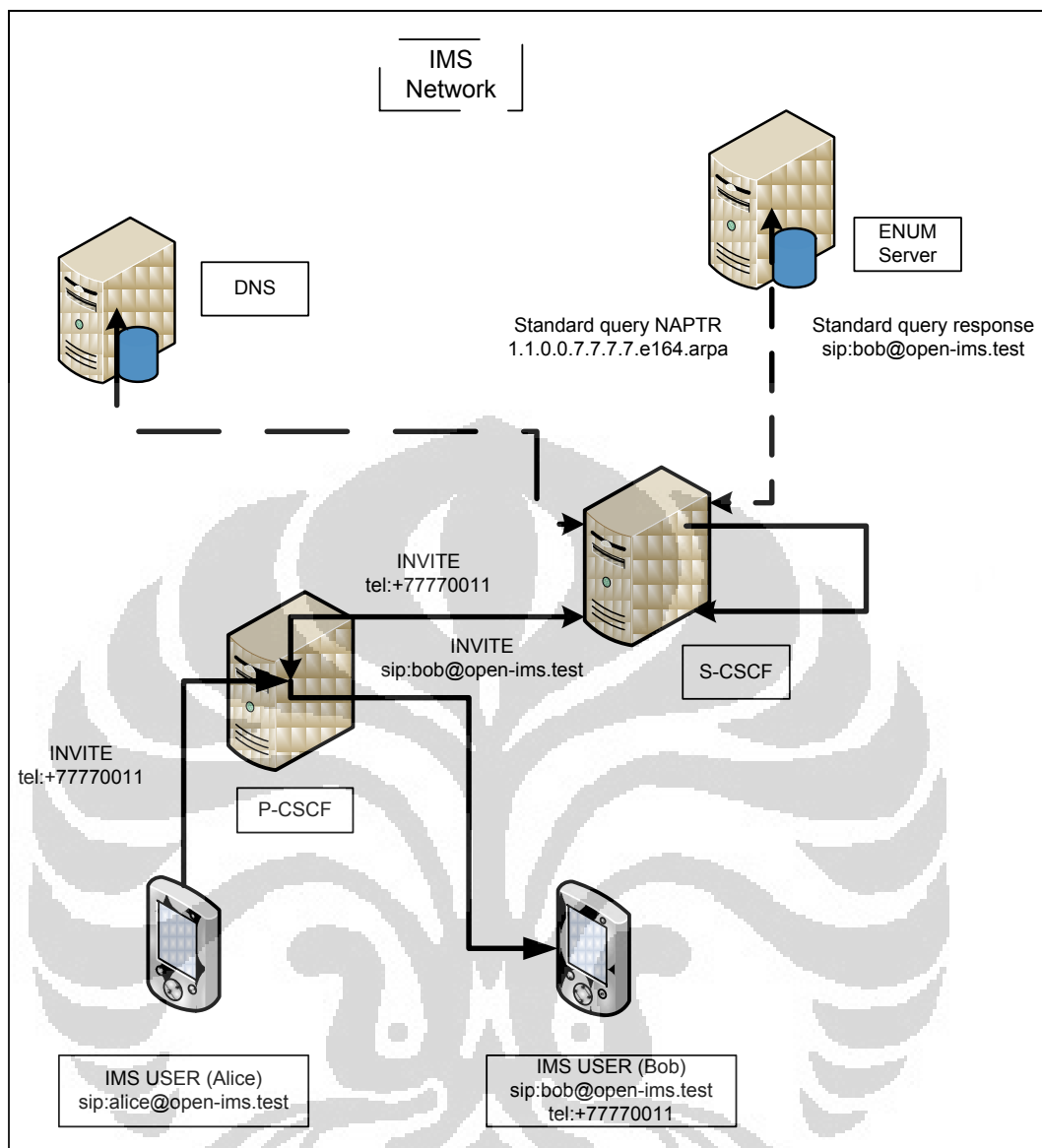
Gambar 3.1 Skenario Penggunaan Umum ENUM Server

Ketika suatu pengguna pada jaringan IMS akan menghubungi pengguna lain (PSTN) yang memiliki identitas berformat tel URI, maka setelah mendapatkan pesan SIP INVITE, S-CSCF akan mengakses ENUM Server untuk mencari alamat SIP pengganti dari alamat tel URI tersebut. Lalu, S-CSCF akan mengirimkan pesan SIP dengan alamat tujuan berupa SIP

pengganti tersebut kepada penerima di jaringan PSTN tersebut, dalam hal ini adalah PSTN *gateway*. Lalu, PSTN *gateway* akan menghubungi pengguna dengan alamat SIP tujuannya tersebut seperti pada gambar 3.1.

Ketika pesan SIP telah sampai pada PSTN *gateway* maka pengolahan pesan akan menjadi tanggung jawab dari PSTN *gateway* tersebut atau titik kontak pertama di domain yang lain. Oleh karena itu, proses yang dijalani setelah berpindah lingkup jaringan, akan mengalami penyesuaian dengan yang berlaku pada jaringan tersebut. Penyesuaian ini akan membuat berbedanya proses keseluruhan secara signifikan antara panggilan yang tidak menggunakan ENUM Server (langsung menggunakan alamat SIP URI), dengan yang menggunakan ENUM Server (menggunakan alamat tel URI), sehingga tujuan pengamatan perbandingan proses penggunaan kedua alamat tidak dapat dilakukan dengan baik. Untuk dapat mengamati proses-proses yang ada saat menggunakan ENUM Server murni didalam jaringan IMS, perlu dilakukan penentuan rute hasil akses ENUM Server agar kembali kedalam jaringan IMS. Hal ini dilakukan agar setiap proses yang dialami ketika menggunakan ENUM Server dan tidak menggunakan ENUM Server dapat dibandingkan dengan jelas, karena mengalami proses-proses pembentukan panggilan yang sama secara umum.

Ketika seorang pengguna IMS ingin menghubungi pengguna lain di dalam IMS yang mempunyai identitas berupa tel URI, maka pengguna pertama kali akan mengirimkan pesan SIP INVITE ke P-CSCF. Kemudian S-CSCF akan mengakses ENUM Server untuk mencari tahu, alamat SIP pengganti dari identitas tel URI tujuan. Ketika telah mendapatkannya, maka S-CSCF akan mengolahnya sebagai pesan SIP INVITE baru dengan alamat SIP pengganti sebagai tujuan. Setelah itu, pengguna akan tersambung seperti panggilan pada jaringan IMS pada umumnya. Skenario ini dapat dilihat pada gambar 3.2.



Gambar 3.2. Skenario Penggunaan ENUM Server dalam Jaringan IMS

### 3.2. Penggunaan Modul ENUM dalam OpenIMSCore

Modul ENUM dalam OpenIMSCore adalah modul yang berasal dari SIP Express Router (SER) yang bersifat *open source*, sehingga dapat digunakan secara bebas, asalkan tetap mencantumkan pembuatnya. Modul ENUM mengimplementasikan fungsi *enum\_query* yang membuat sebuah permintaan akses ENUM berdasarkan dari bagian dari *Request URI* dari user yang sedang berlangsung. Fungsi ini menganggap bahwa bagian dari user terdiri atas bentuk nomor telepon berformat internasional, yaitu dengan bentuk tanda plus



(+) ditambah dengan digit desimal, dengan jumlah minimal dua dan maksimal 15 nomor. Jika bertemu dengan format nomor seperti ini, modul ENUM akan membalik urutan nomor, dan memisahkannya dengan titik. Selain itu, juga ditambahkan akhiran domain, yang mempunyai domain default "e164.arpa". Sebagai contoh, jika nomor user adalah +35831234567, nama domain akan menjadi "7.6.5.4.3.2.1.3.8.5.3.e164.arpa".

Setelah membentuk sebuah nama domain, ENUM akan mengakses ENUM server yang berupa DNS server untuk mengakses query dari bentuk NAPTR. Dari respon yang tersedia, fungsi *enum\_query* akan memilih data, dengan tanda yang mempunyai karakter "u", dan dengan tanda layanan "e2u+[service:]sip" atau "e2u+type[:subtype][+type[:subtype]...]", dan yang mempunyai data di kolom pengganti.

Setelah itu, *enum\_query* akan memilih data NAPTR berdasarkan order dan preferencinya. Setelah di pilih, fungsi *enum\_query* akan mengganti *Request* URI yang lama dengan data pengganti yang paling sesuai dari Data NAPTR.

Ketika menggunakan fungsi *enum\_query* tanpa parameter lain, fungsi akan mencari data simpanan NAPTR dengan tipe layanan "e2u+sip" dalam enum. Ketika digunakan parameter tunggal, parameter ini akan digunakan dalam pohon urutan enum. Ketika digunakan dua parameter, fungsinya tergantung kepada huruf pertama pada parameter kedua. Ketika huruf pertama adalah bukan tanda "+", maka parameter kedua akan digunakan untuk mencari data NAPTR dengan tipe layanan "e2u+parameter:sip". Ketika parameter kedua dimulai dengan tanda +, maka Enum akan mencari data NAPTR gabungan (contoh : "e2u+voice:sip+video:sip"), dan mencari untuk berbagai macam layanan dalam satu pencarian. Berbagai macam layanan harus dipisahkan dengan tanda +.

### 3.2.1. Fungsi *enum\_query*

Terdapat tiga jenis fungsi *enum\_query*, yaitu *enum\_query()*, *enum\_query ("suffix")*, *enum\_query ("suffix", "service")*. Fungsi-fungsi ini melakukan pencarian query enum dan menuliskan kembali permintaan URI dengan hasil yang didapat dari data. Arti dari parameter-parameter yang

terdapat dalam fungsi *enum\_query* adalah sebagai berikut.

- *Suffix* : Akhiran yang akan ditambahkan kedalam nama domain.
- *Service* : Layanan yang akan digunakan dalam tabel layanan.

Berikut ini adalah beberapa contoh pengaturan penggunaan fungsi *enum\_query* agar dapat digunakan sesuai dengan kebutuhan.

1. Mencari dalam suatu domain khusus

Untuk mencari data dalam suatu domain penyedia Enum khusus, domain tersebut harus didefinisikan didalam fungsi *enum\_query* seperti contoh berikut. Mencari didalam domain *frenum.org*.

```
enum_query("frenum.org.");
```

2. Mencari didalam domain default.

Untuk mencari di dalam domain yang sudah biasa digunakan, *enum\_query* tidak perlu didefinisikan kembali. Enum akan mencari data didalam rantai DNS yang didefinisikan didalam parameter saat modul di *load*. Bentuk layanan pada yang dicari adalah bersifat tetap, yaitu *e2u+sip*.

```
enum_query();
```

3. Mencari sebuah layanan didalam suatu domain.

Untuk mencari data dengan layanan yang spesifik diinginkan didalam suatu domain, diperlukan format penulisan fungsi *enum\_query* seperti berikut. Contohnya adalah mencari layanan suara didalam domain *e164.arpa*.

```
enum_query ("e164.arpa.", "voice");
```

4. Mencari lebih dari satu layanan didalam suatu domain.

Serupa dengan pencarian data layanan dengan sebuah layanan, untuk mencari data layanan lebih dari satu dalam suatu domain, diperlukan penulisan fungsi *enum* yang khusus. Contoh berikut adalah mencari layanan sip, suara dan video didalam domain *e164.org*.

```
enum_query("e164.arpa.", "+sip+voice:sip+video:sip");
```

### 3.2.2. Proses Load Modul ENUM dalam OpenIMSCore

Saat I-CSCF dan S-CSCF di jalankan, perintah untuk memanggil modul enum yang telah dituliskan beserta parameternya akan berjalan. Modul enum yang telah dipanggil akan mulai aktif saat itu juga. Setelah itu, modul enum akan mencari parameter-parameter yang telah diatur di bawah perintah pemanggilan module. Parameter yang pertama adalah parameter untuk pengantian tel URI dengan format SIP URI, dan yang kedua adalah pengantian dengan TEL URI. Untuk pengantian tel URI dengan SIP URI, parameter-parameter tersebut adalah kecocokan nama modul “enum”, perintah adanya *domain\_suffix* (domain yang ada di akhir), serta nama domain yang telah diset. Sementara untuk pengantian dengan TEL URI, parameternya adalah kecocokan nama modul “enum”, dan “tel\_uri\_params”.

### 3.3. Konfigurasi ICSCF dan SCSCF

Untuk dapat menggunakan fungsi ENUM didalam OpenIMSCore, diperlukan beberapa konfigurasi yang ada didalam *Interrogating-CSCF* (I-CSCF) dan *Serving-CSCF* (S-CSCF).

Konfigurasi parameter dengan menambahkan parameter di file konfigurasi *icscf.cfg* dan *scscf.cfg*

- Tambahkan parameter pada modul enum yang ada di OpenIMSCore. Di dalam parameter tersebut, tuliskan nama domain yang akan dicari saat S-CSCF mengirimkan *NAPTR Request* .  

```
modparam("enum", "domain_suffix", "e164.arpa.")
modparam("enum", "tel_uri_params", " ") #;npdi
```
- Rubah rute *PSTN\_handling* pada konfigurasi I-CSCF dan S-CSCF menuju alamat IP dan port tujuan . Alamat IP dan port tujuan yang diatur didalam I-CSCF dan S-CSCF tergantung komponen apa yang akan menangani hasil translasi dari module enum. Karena pada penelitian ini, hasil akses ENUM akan diarahkan kedalam jaringan IMS kembali, maka alamat IP dan port tujuan diisi oleh alamat IP dan port S-CSCF.

### 3.4. Konfigurasi ENUM Server

ENUM server bekerja dengan prinsip kerja DNS Server. Oleh karena itu, yang dibutuhkan adalah konfigurasi ENUM, agar fungsi pencarian alamat dapat berjalan sesuai dengan prinsip kerja ENUM. Instalasi *software* DNS Server telah dilakukan pada instalasi OpenIMSCore. Jika belum, yang harus dilakukan adalah instalasi DNS Server dengan Software BIND9.

Ada beberapa tahapan dalam membuat ENUM server.

- Menambahkan sebuah perintah dalam folder akses utama DNS dengan BIND yaitu *named.conf*. Konfigurasi yang dibuat adalah konfigurasi untuk membuat sebuah DNS server baru yang khusus digunakan untuk ENUM server.
- Membuat file berisi DNS /ENUM server baru dan mengisinya dengan nama zona yang digunakan oleh ENUM server.
- Membuat sebuah file *database* query ENUM dan mengisi dengan semua query ENUM server yang diperlukan.
- Tambahkan data-data yang menjelaskan isi dari ENUM server, sesuai dengan pengaturan yang ada seperti gambar 3.1 berikut.

```

$TTL 86400
e164.arpa.      IN      SOA  ns.e164.arpa. root.e164.arpa. (
                2011061104 ; Serial no., based on date
                21600 ; Refresh after 6 hours
                3600 ; Retry after 1 hour
                604800 ; Expire after 7 days
                3600 ; Minimum TTL of 1 hour)
e164.arpa.     43200 IN      NS   ns.e164.arpa.
;
ns.e164.arpa.  43200 IN      A    10.10.0.197
1.1.0.0.7.7.7.7.e164.arpa. NAPTR 10 100 "u" "E2U+sjp" "!^.*$!sip:bob@open-ims.test!".
2.2.0.0.7.7.7.7.e164.arpa. NAPTR 10 101 "u" "E2U+sjp" "!^.*$!sip:faisal@open-ims.test!".
3.3.0.0.7.7.7.7.e164.arpa. NAPTR 10 102 "u" "E2U+sjp" "!^.*$!sip:rosa@open-ims.test!".
4.4.0.0.7.7.7.7.e164.arpa. NAPTR 10 103 "u" "E2U+sjp" "!^.*$!sip:chandra@open-ims.test!".

```

Gambar 3.3. Contoh Pengaturan Data Pada ENUM Server

Simpan.

- Tambahkan konfigurasi *logging* dalam *named.conf.options* agar dapat lebih mudah melihat kesalahan-kesalahan dalam pengaturan *query log*. Gambar 3.2 adalah contoh konfigurasi *logging* ENUM server

```
#logging {
# channel query-log {
# file "/var/log/query-log" versions 2 size 10m;
# print-time yes;
# print-category yes;
# };
# category queries { query-log; };
#};
```

Gambar 3.4. Konfigurasi Logging

### 3.5. Pengujian Konfigurasi ENUM Server dari Komputer Server

Lakukan pengujian apakah ENUM Server telah berjalan dengan baik dengan tools *dig* (Domain Information Groper). *Dig* adalah alat berupa sekumpulan perintah dalam administrasi jaringan untuk mengecek DNS server apakah menghasilkan data DNS yang seharusnya atau tidak. *Dig* sangat berguna untuk pemecahan masalah jaringan atau tujuan pendidikan.

```
#dig @ 10.10.0.197 1.1.0.0.7.7.7.7.e164.arpa -t NAPTR
```

Jika tidak terjadi kesalahan pengaturan dalam ENUM Server, maka hasil yang akan didapat adalah seperti gambar 3.1.

```
faisal@ubuntu:~$ dig @10.10.0.197 -t NAPTR 1.1.0.0.7.7.7.7.e164.arpa

; <<>> DiG 9.7.0-P1 <<>> @10.10.0.197 -t NAPTR 1.1.0.0.7.7.7.7.e164.arpa
; (1 server found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 65009
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;1.1.0.0.7.7.7.7.e164.arpa. IN NAPTR

;; ANSWER SECTION:
1.1.0.0.7.7.7.7.e164.arpa. 604800 IN NAPTR 10 100 "u"*E2U+sip* "!\^.*$!sip:bob@open-ims.test!".

;; AUTHORITY SECTION:
e164.arpa. 604800 IN NS ns.e164.arpa.

;; ADDITIONAL SECTION:
ns.e164.arpa. 604800 IN A 10.10.0.197

;; Query time: 1 msec
;; SERVER: 10.10.0.197#53(10.10.0.197)
;; WHEN: Sat Jun 11 10:43:41 2011
;; MSG SIZE rcvd: 137
```

Gambar 3.5 Pengujian Konfigurasi ENUM Server dari Komputer Server

Pada hasil dig terlihat bahwa, Enum Server telah dapat memberikan jawaban atas permintaan yang diminta oleh *software* dig. Yang diminta adalah hasil dari pertanyaan (*question section*), yang meminta hasil dari nama domain *1.1.0.0.7.7.7.7.e164.arpa* dengan jenis *query* NAPTR.

### 3.6. Pengujian Konfigurasi ENUM Server dari Komputer *Client*

Pengujian konfigurasi ENUM Server juga dilakukan dari komputer *Client* agar benar-benar dapat dibuktikan bahwa ENUM server dapat diakses oleh *Client*. Pengujian dilakukan dengan cara melakukan *ping* kepada nama domain yang ada, dan melakukan *dig* dari komputer *Client*. Jika semua pengaturan telah tepat, baik di komputer *client* ataupun komputer server, maka proses *ping* dan *dig* akan mengeluarkan hasil yang tepat, seperti gambar-gambar dibawah ini.

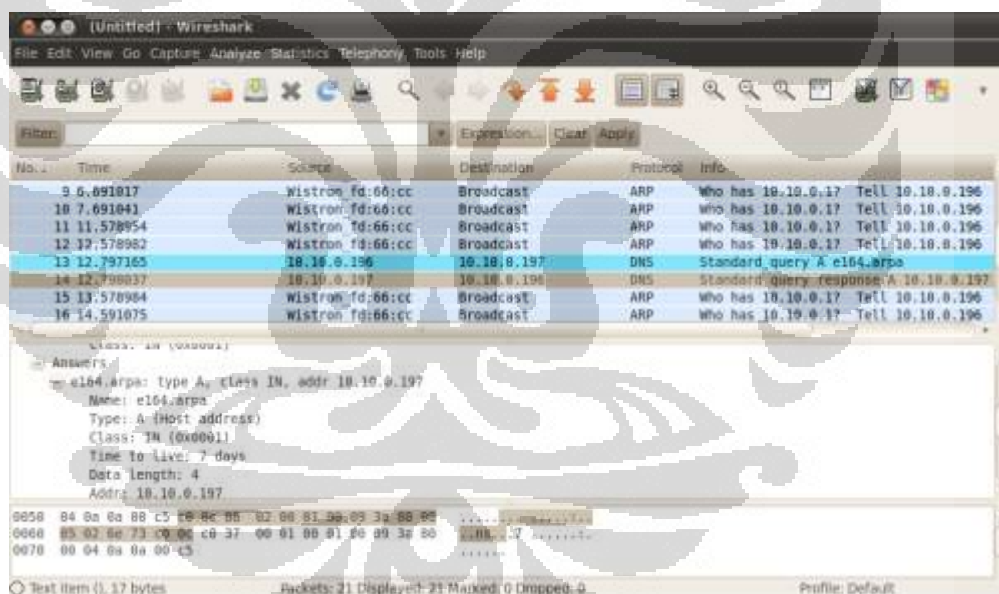
Pengujian konfigurasi dengan menggunakan ping e164.arpa sangat dibutuhkan untuk menguji apakah komputer sudah mempunyai DNS server dengan domain e164.arpa dengan baik. Jika ping e164 berhasil, berarti DNS

server telah dapat mengenali domain tersebut menuju suatu alamat IP seperti gambar 3.4. Ping e164.arpa dari Komputer *Clientt* (IP *Client* = 10.10.0.196)

```
faisal@faisal-laptop:~$ ping e164.arpa
PING e164.arpa (10.10.0.197) 56(84) bytes of data.
64 bytes from e164.arpa (10.10.0.197): icmp_seq=1 ttl=64 time=0.076 ms
64 bytes from e164.arpa (10.10.0.197): icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from e164.arpa (10.10.0.197): icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from e164.arpa (10.10.0.197): icmp_seq=4 ttl=64 time=0.053 ms
^C
-- e164.arpa ping statistics --
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.053/0.060/0.076/0.013 ms
```

Gambar 3.6 . Hasil Ping domain e164.arpa dari Terminal Komputer *Client*

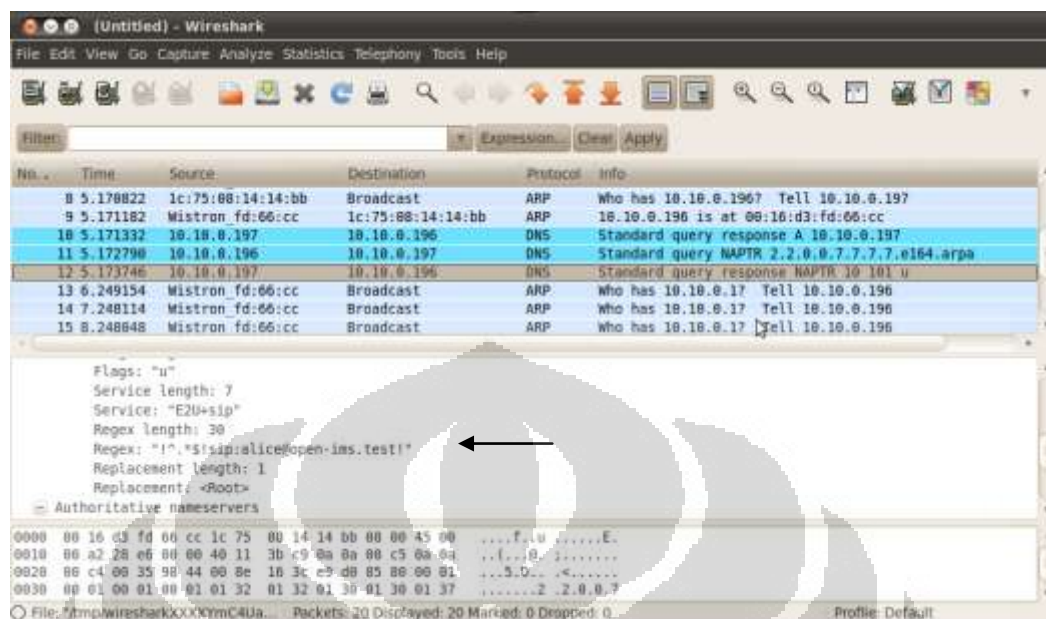
Jika dilihat dengan menggunakan Wireshark, maka akan terlihat bahwa terjadi permintaan *query* dari computer kepada ENUM Server, untuk mencari alamat IP pengganti dari domain e164.arpa seperti yang terlihat pada gambar 3.5.



Gambar 3.7. Pengujian Konfigurasi dengan Ping e164.arpa Menggunakan *Wireshark*

Cara pengujian selanjutnya adalah menggunakan peralatan DIG. Jika konfigurasi berhasil, maka DIG akan memunculkan jawaban pada terminal dan dapat dilihat lebih detail menggunakan *Wireshark*, seperti pada gambar 3.6.

`#dig @e164.arpa 2.2.0.0.7.7.7.7.e164.arpa -t NAPTR`



Gambar 3.8 Pengujian Konfigurasi ENUM Server dengan Dig Menggunakan *Wireshark*

Dari hasil dig yang didapat dari terminal dan pengamatan *Wireshark*, maka telah didapatkan hasil keluaran berupa SIP pengganti pada kolom Regex, yaitu [sip:alice@open-ims.test](mailto:sip:alice@open-ims.test). Oleh karena itu, dapat disimpulkan bahwa ENUM Server telah berjalan dengan baik.



## BAB 4

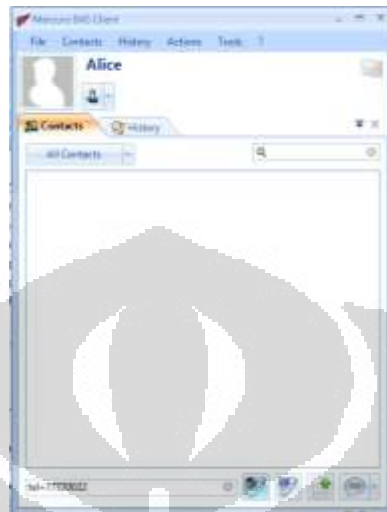
### HASIL PENGUJIAN DAN ANALISIS

#### 4.1. Prosedur Pengambilan Data

Pengambilan data dilakukan dengan menggunakan Wireshark *Network Protocol Analyzer* yang dijalankan pada server OpenIMSCore dan ENUM Server. Pada analisis ini, diamati bagaimana perbandingan tahapan proses pengolahan panggilan dan lama proses antara panggilan dengan menggunakan format tel URI dengan proses panggilan dengan menggunakan format SIP URI. Selain itu akan diamati lama proses pengolahan permintaan data sampai diberikan jawaban data oleh Enum Server.

1. Proses pengolahan panggilan akan dilakukan dengan cara mendaftarkan sebuah IMS *Client* dengan menggunakan suatu identitas yang telah didaftarkan sebelumnya di server IMS yaitu [sip:alice@open-ims.test](mailto:sip:alice@open-ims.test) . Pada proses pengolahan panggilan dengan menggunakan format SIP URI, *client* [sip:alice@open-ims.test](mailto:sip:alice@open-ims.test) melakukan panggilan langsung ke format SIP URI , [bob@open-ims.test](mailto:bob@open-ims.test) . Dari panggilan ini, akan diamati proses yang terjadi dari alur pesan yang didapatkan dengan menggunakan Wireshark, serta lama proses yang dijalani dari mulai tombol *dial* ditekan (pesan INVITE dikirim) sampai didapatkan pesan 200 OK pada *Client*.
2. Pada proses pengolahan panggilan dengan menggunakan format TEL URI, *client* yang telah terdaftar [sip:alice@open-ims.test](mailto:sip:alice@open-ims.test) akan melakukan panggilan ke format TEL URI, yaitu <tel:+77770011>. Pada proses ini juga akan diamati proses yang terjadi dari alur pesan INVITE, sampai didapatkan pesan 200 OK pada *Client* .
3. Waktu proses panggilan diamati sejak pesan INVITE pertama dikirim oleh pemanggil sampai didapatkan pesan 200 OK diterima oleh pemanggil, baik panggilan yang melalui ENUM server maupun tidak melalui ENUM server.
4. Waktu proses yang terjadi pada ENUM server diamati dan di hitung dari mulai saat menerima permintaan akses data NAPTR sampai memberikan jawaban data NAPTR.

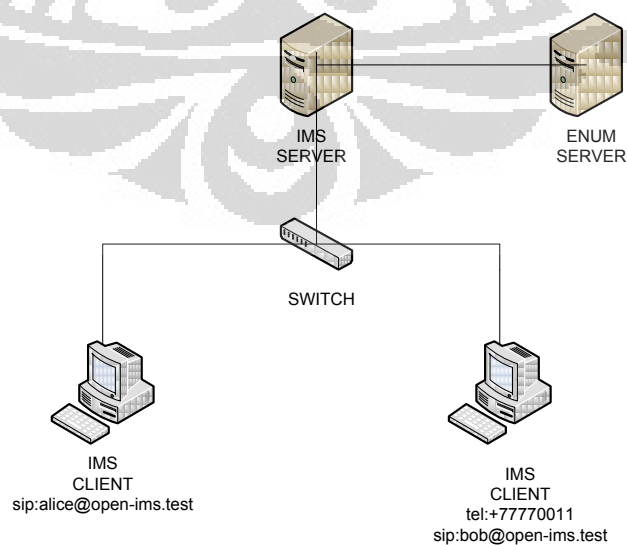
IMS *Client* yang digunakan dalam skripsi ini adalah IMS *Client* Mercuro Bronze, karena telah mampu mendukung penggunaan format tel URI dalam berkomunikasi.



Gambar 4.1. Mercuro Bronze IMS Client

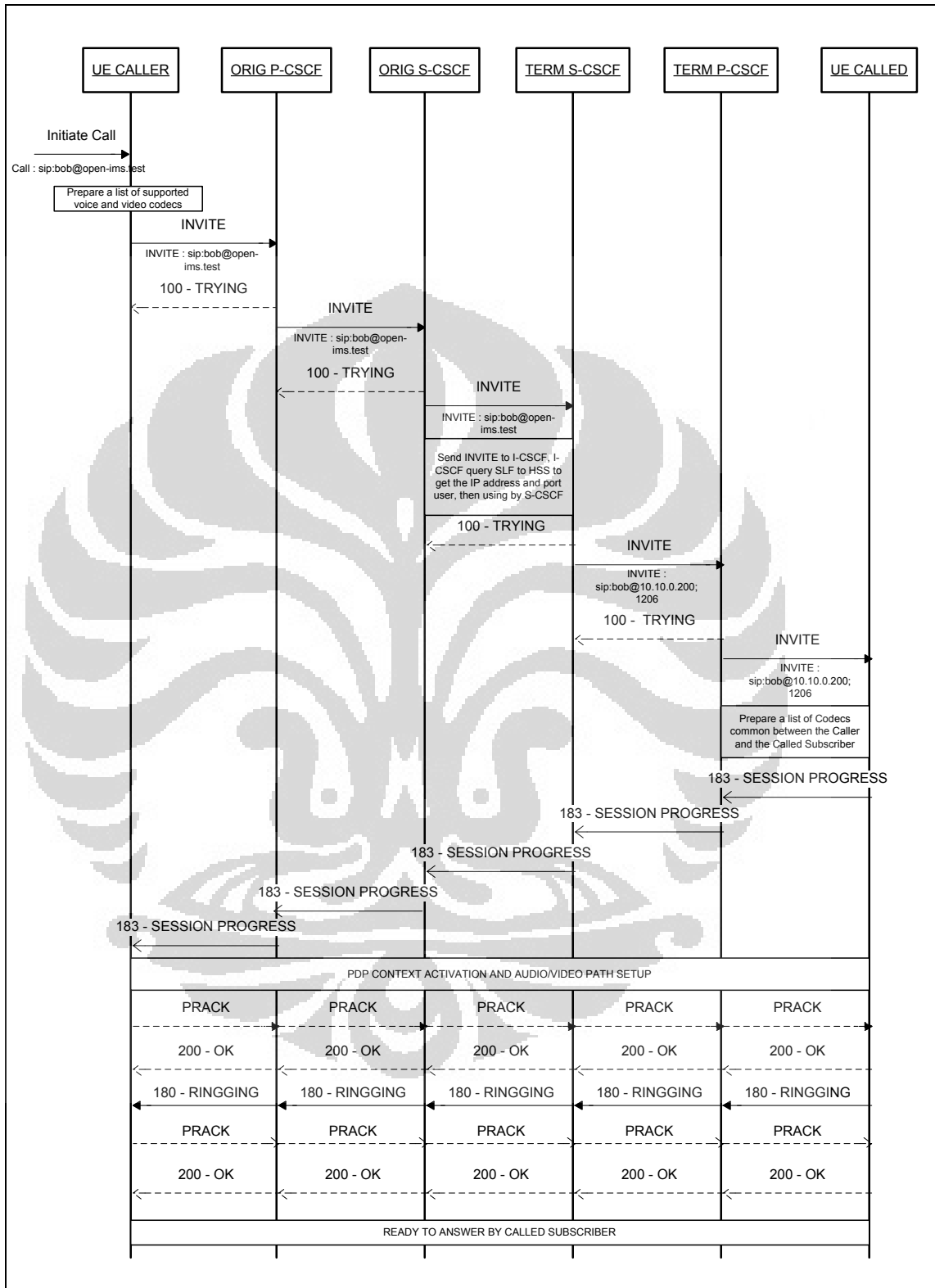
#### 4.2. Topologi Jaringan Percobaan

Pada uji coba ini, digunakan topologi seperti gambar 4.2. Server OpenIMSCore mempunyai IP 10.10.0.196, dengan masing-masing komponen IMS di port yang berbeda. ENUM server mempunyai IP 10.10.0.198, IMS *Client* Alice mempunyai IP 10.10.0.197, dan IMS *Client* Bob memiliki IP 10.10.0.200.



Gambar 4.2 Topologi Jaringan Percobaan ENUM Server - OpenIMSCore

### 4.3. Proses Kerja Panggilan Menggunakan SIP URI



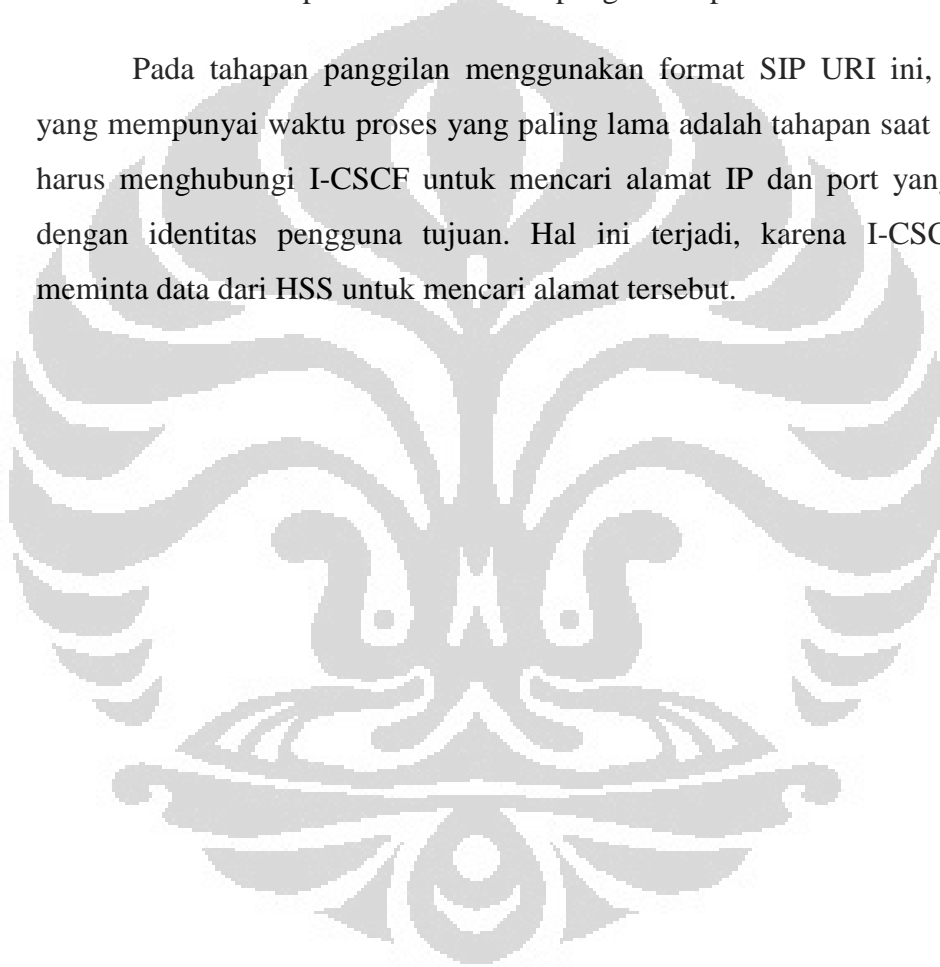
Gambar 4.3 Alur Pesan Pemanggilan Menggunakan SIP URI

Proses kerja panggilan menggunakan SIP URI sebagai format tujuan adalah proses kerja standar yang berlaku pada panggilan VoIP pada IMS. Proses kerja panggilan tersebut dapat dilihat pada gambar 4.3. Pada awalnya, seorang dengan sebuah pengguna yang telah terdaftar dengan identitas [sip:alice@open-ims.test](mailto:sip:alice@open-ims.test) (Alice) memanggil pengguna lain dengan alamat [sip:bob@open-ims.test](mailto:sip:bob@open-ims.test) (Bob). Saat panggilan mulai dilakukan, pengguna akan mengirimkan pesan SIP INVITE [sip:bob@open-ims.test](mailto:sip:bob@open-ims.test) kepada P-CSCF. P-CSCF menerima pesan tersebut dan mengirimkan pesan Status 100-Trying kepada pengguna pemanggil Alice sebagai tanda bahwa pesan telah diterima dengan baik dan akan segera diproses. Setelah mengirimkan pesan Status 100-trying, P-CSCF mengirimkan pesan *Request* INVITE [sip:bob@open-ims.test](mailto:sip:bob@open-ims.test) kepada S-CSCF. S-CSCF mengirimkan pesan Status 100-Trying kepada P-CSCF sebagai tanda pesan telah diterima dan akan segera di proses. S-CSCF akan menghubungi I-CSCF agar I-CSCF mengkontak SLF (*Subscriber Location Function*) untuk mencari lokasi alamat IP dan port pengguna dimana Bob berada. Hasil alamat IP dan Port tersebut akan digunakan S-CSCF untuk mengirimkan kembali pesan INVITE kepada P-CSCF dengan format yang telah berisi alamat IP dan port pengguna bob, yaitu [sip:bob@10.10.0.200:1260](mailto:sip:bob@10.10.0.200:1260), dengan transport berjenis protokol UDP. Ketika menerima pesan tersebut, P-CSCF akan melakukan konfirmasi penerimaan dengan pesan 100-Trying.

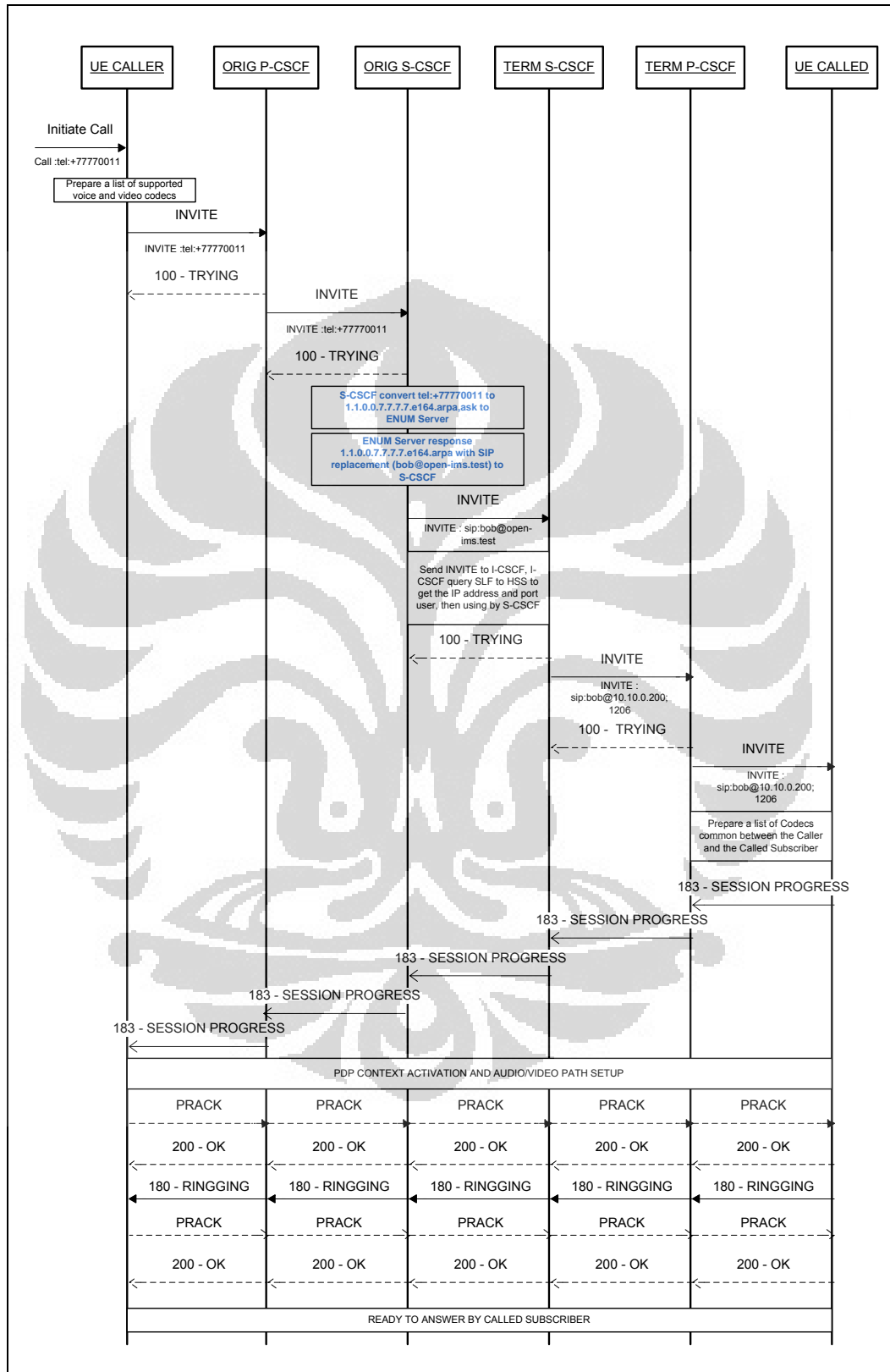
P-CSCF selanjutnya akan mengirimkan pesan INVITE langsung kepada alamat IP, port dan *username* yang telah diketahui. Sesaat setelah menerima pesan INVITE tersebut, pengguna akan melakukan penyesuaian Codec dengan yang dikirimkan pemanggil. Pengguna yang telah menerima pesan tersebut akan melakukan konfirmasi dengan mengirimkan pesan 183 Session Progress kepada P-CSCF sebagai tanda sesi komunikasi sedang dipersiapkan. P-CSCF akan mengirimkan pesan 183 Session Progress kepada S-CSCF untuk di lakukan pencatatan oleh S-CSCF. Setelah dilakukan pencatatan oleh S-CSCF, maka pesan 183 Session Progress akan dikirimkan kembali ke P-CSCF lalu dikirimkan ke *client*. Ketika *client* telah menerima pesan 183 Session Progress, maka diaktifkannya PDP context dan pengaturan

jalur video atau suara. Masing-masing *client* akan mengaktifkan PDP context yang akan berperan pada pengalokasian sumber daya yang harus dipenuhi untuk menyesuaikan dengan *Quality of Service (QoS)* yang telah ditentukan. *Client* menerima pesan tersebut dan memberikan pesan balasan dengan PRACK. Pesan PRACK akan diteruskan ke S-CSCF untuk dilakukan pencatatan. Pesan PRACK yang telah dicatat akan dikembalikan ke PCSCF, dan pesan 200 OK akan dikirimkan oleh *Client*, sebagai tanda bahwa *client* telah benar-benar siap untuk melakukan pengiriman pesan suara.

Pada tahapan panggilan menggunakan format SIP URI ini, tahapan yang mempunyai waktu proses yang paling lama adalah tahapan saat S-CSCF harus menghubungi I-CSCF untuk mencari alamat IP dan port yang sesuai dengan identitas pengguna tujuan. Hal ini terjadi, karena I-CSCF akan meminta data dari HSS untuk mencari alamat tersebut.



### 4.4. Proses Kerja Panggilan Menggunakan TEL URI



Gambar 4.4 Alur Pesan Pemanggilan Menggunakan tel URI

Pada panggilan menggunakan TEL URI, pada awalnya *Client* akan mengirimkan *Request* INVITE <tel:+77770011> kepada P-CSCF. P-CSCF akan menerima permintaan tersebut, dan mengirimkan pesan 100-Trying kepada *client* sebagai tanda *request* telah diterima, dan akan segera diproses. P-CSCF akan mengirimkan *request* INVITE <tel:+77770011> tersebut ke S-CSCF. S-CSCF yang memiliki algoritma ENUM, akan melakukan pengolahan dari format tel URI, menjadi format ENUM *request* yang ada. Pada percobaan ini, S-CSCF akan mengolah masukan format tel URI <tel:+77770011> menjadi format ENUM *request* 1.1.0.0.7.7.7.7.e164.arpa . S-CSCF akan mengakses data pada ENUM server, dengan pertanyaan berupa permintaan data NAPTR dengan format ENUM *request* tersebut. ENUM server akan menjawab permintaan data dengan alamat pengganti yang memiliki format SIP URI. Pada *database* ENUM server, <tel:+77770011> memiliki alamat pengganti berupa <sip:bob@open-ims.test> (Bob). Selanjutnya, S-CSCF akan mengirimkan 100-Trying sebagai tanda bahwa S-CSCF telah menemukan alamat sip pengganti. Seperti pada proses menggunakan SIP URI, S-CSCF akan menghubungi I-CSCF agar I-CSCF mengkontak SLF (*Subscriber Location Function*) untuk mencari lokasi alamat IP dan port pengguna dimana Bob berada. Hasil alamat IP dan Port tersebut akan digunakan S-CSCF untuk mengirimkan kembali pesan INVITE kepada P-CSCF dengan format yang telah berisi alamat IP dan port pengguna bob, yaitu <sip:bob@10.10.0.200:1260>, dengan transport berjenis protokol UDP. Setelah didapatkan dimana posisi pengguna dengan identitas tersebut, maka S-CSCF akan memberitahu kembali P-CSCF agar dapat membangun jalur data untuk membangun panggilan. P-CSCF akan menerima pesan tersebut dan memberikan jawaban 100-Trying sebagai tanda pesan telah diterima dan akan segera diproses lebih lanjut..

P-CSCF yang telah mempunyai alamat IP dan port dari “Bob”, akan langsung menghubungi pengguna “Bob” dengan alamat IP dan port yang ada dengan pesan INVITE. Sesaat setelah menerima pesan INVITE tersebut, akan disiapkanlah Codec yang sesuai antara “Bob” dengan pemanggil yaitu “Alice”. Pengguna “Bob” menjawab dengan memberikan pesan SIP 183

Session Progress, kepada P-CSCF. P-CSCF akan menerima pesan tersebut dan meneruskannya kepada S-CSCF untuk dilakukan pencatatan. S-CSCF akan melakukan pencatatan dan mengirimkan kembali ke P-CSCF. P-CSCF akan mengirimkan pesan 183 Session Progress tersebut kepada pengguna pemanggil, yaitu "Alice". Setelah "Alice" menerima pesan 183 Session Progress, PDP Context akan segera diaktifkan dan pengaturan jalur video dan suara segera dilakukan. PDP Context ini diaktifkan oleh masing-masing pengguna, agar dilakukannya alokasi sumber daya data untuk memenuhi QoS yang dimiliki atau dipersyaratkan kedua pengguna tersebut.

Pengguna "Alice" akan mengirimkan pesan PRACK kepada P-CSCF sebagai tanda telah mengetahui panggilannya akan berhasil. Pesan PRACK akan diteruskan oleh P-CSCF ke S-CSCF, dan dikembalikan ke P-CSCF sebagai tanda telah dicatat oleh S-CSCF. P-CSCF akan mengirimkan pesan PRACK tersebut ke sisi pengguna "Bob". Bob akan menerima pesan PRACK dan membalas dengan Pesan 200 OK. Pesan ini akan dikirimkan ke P-CSCF, lalu ke S-CSCF, lalu dikembalikan kembali ke P-CSCF. Pengguna "Bob" akan mulai mengirimkan pesan 180 Ringing sebagai tanda telah berdering dan benar-benar siap memulai panggilan. Setelah pesan 180 Ringing diterima oleh pemanggil "Alice", akan diberikan konfirmasi penerimaan pesan oleh Alice berupa pesan Prack. Setelah itu pengguna "Bob" juga akan mengirimkan pesan SIP Status : 200 OK sampai ke Alice sebagai tanda panggilan benar-benar siap diangkat.

Pada proses panggilan menggunakan tel URI ini, didapatkan tahapan tambahan yaitu tahapan akses ENUM Server oleh S-CSCF. S-CSCF akan mengirimkan pertanyaan dengan format ENUM *Request* kepada ENUM Server, lalu ENUM Server akan memberikan alamat SIP pengganti jika terdapat kecocokan antara *request* dengan data yang ada didalam ENUM Server. Tahapan ini adalah tahapan yang harus dilakukan untuk mencari alamat SIP pengganti, sehingga tahapan ini dapat dipastikan mempengaruhi lamanya proses pembangunan panggilan.



Selain akses ENUM Server oleh S-CSCF, pada tahapan yang S-CSCF menerima pesan INVITE, didapatkan bahwa tahapan tersebut memiliki waktu proses yang lebih lama dibandingkan pada proses menggunakan SIP URI. Hal ini terjadi karena S-CSCF harus melakukan proses perubahan format terlebih dahulu sebelum mengakses ENUM Server. S-CSCF harus mengubah format tel URI menjadi format ENUM *request*. Perubahan ini dilakukan dengan menggunakan algoritma yang ada didalam S-CSCF.

#### 4.5. Perbandingan Lama Waktu Proses Kerja

Pada proses panggilan menggunakan SIP URI, dari data yang didapatkan dari percobaan, didapatkan lama waktu proses dari mulai INVITE [sip:bob@open-ims.test](mailto:sip:bob@open-ims.test) sampai pada didapaknya Status 200 OK rata-rata adalah sebesar 0.1071274 detik. Lama waktu proses ini disebabkan oleh hubungan antara *user* dengan elemen-elemen di dalam OpenIMSCore, dan hubungan antar elemen di dalam OpenIMSCore.

Tabel 4.1. Data Lama Waktu Kerja Pemanggilan Format SIP URI

No. Percobaan	Waktu Proses (s)	No. Percobaan	Waktu Proses (s)
1	0.105022	11	0.124342
2	0.098791	12	0.098485
3	0.112692	13	0.122427
4	0.122022	14	0.102563
5	0.108255	15	0.103312
6	0.107374	16	0.101257
7	0.093321	17	0.109596
8	0.097633	18	0.113771
9	0.107756	19	0.096459
10	0.112295	20	0.105174

<b>Rata-rata (s)</b>	<b>0.1071274</b>
----------------------	------------------

Dari rata-rata waktu proses pembentukan panggilan pada tabel 4.1, dapat dibandingkan dengan performa respon panggilan. Performa respon

panggilan VoIP menyatakan bahwa nilai rata-rata respon seharusnya bernilai maksimal 205-218 ms. Nilai yang didapatkan dari percobaan adalah 0.1071274 s atau 107,1 ms. Oleh karena itu dapat dinyatakan bahwa proses panggilan dengan menggunakan SIP URI sesuai di dalam standar yang ada.

Pada proses panggilan menggunakan tel URI, dari data yang didapatkan dari percobaan, didapatkan lama waktu proses dari mulai INVITE : <tel:+77770011> sampai pada Status 200 OK, rata-rata waktu proses adalah sebesar 0.1332041 detik.

Tabel 4.2. Data Lama Waktu Kerja Pemanggilan Format TEL URI

No. Percobaan	Waktu Proses (s)	No. Percobaan	Waktu Proses (s)
1	0.17245	11	0.136567
2	0.153159	12	0.121553
3	0.119784	13	0.146299
4	0.12596	14	0.132162
5	0.127729	15	0.179866
6	0.121277	16	0.104339
7	0.112706	17	0.139345
8	0.153292	18	0.130047
9	0.123442	19	0.168601
10	0.122242	20	0.113018

<b>Rata-rata</b>	<b>0.1332041</b>
------------------	------------------

Pada proses panggilan menggunakan SIP URI, didapatkan data rata-rata sebesar data yang didapatkan. Pada proses panggilan dengan menggunakan SIP URI, proses panggilan mempunyai tahapan yang standar panggilan VoIP pada jaringan IMS. Setelah alamat tujuan pengguna ditekan, P-CSCF akan langsung mengirimkan pesan *Request* INVITE berupa alamat SIP yang dikenal pada HSS, sehingga S-CSCF dapat langsung memilihkan alamat IP dan port letak pengguna berada.

Sementara pada proses panggilan dengan menggunakan TEL URI, alamat tujuan pengguna yang ditekan bukanlah alamat yang umum digunakan dalam komunikasi IMS, dan internet secara umum. Oleh karena itu, S-CSCF harus melakukan *request* permintaan alamat SIP yang dimiliki oleh pengguna

yang memiliki identitas TEL URI tersebut. Hal ini berarti, terdapat waktu tambahan yang dibutuhkan untuk mengakses pengguna yang memiliki identitas berupa TEL URI.

Dari data percobaan seperti tertera pada tabel 4.3, jika dibandingkan dengan Respon Pencarian dalam waktu yang cepat, waktu proses pembentukan panggilan melalui ENUM Server masih cukup didalam standard yang ada. Dalam performa respon pencarian menyatakan bahwa rata-rata nilai maksimum respon pencarian adalah 205-218 ms. Data percobaan yang didapat adalah 133.2 ms, sehingga dapat disimpulkan waktu proses yang didapatkan masih tergolong cepat dibandingkan dengan standard respon pencarian dalam VoIP.

Data pemrosesan ENUM Server didapatkan memiliki lama rata-rata sebesar 0.0007364 detik (0.73 ms) untuk satu kali permintaan data ENUM Server. Dengan data ini, dapat disimpulkan bahwa ENUM Server dapat memberikan hasil *query* data dengan cepat.

Tabel 4.3 Data Waktu Proses Akses ENUM Server

No. Percobaan	Waktu Proses (s)	No. Percobaan	Waktu Proses (s)
1	0.000697	11	0.000855
2	0.000689	12	0.000722
3	0.000733	13	0.000724
4	0.000705	14	0.000773
5	0.000704	15	0.000746
6	0.000706	16	0.000624
7	0.00076	17	0.000708
8	0.000781	18	0.000746
9	0.000735	19	0.000776
10	0.000763	20	0.000781

<b>Rata-rata (s)</b>	<b>0.0007364</b>
----------------------	------------------

## **BAB 5**

### **KESIMPULAN**

Dari keseluruhan pembahasan dalam skripsi ini, dapat disimpulkan beberapa hal yang bisa dikaitkan dengan Implementasi ENUM Server pada IMS, yaitu

1. Tahapan yang terjadi ketika menggunakan ENUM Server dalam panggilan adalah meneruskan pesan INVITE tel URI ke PCSCF, PCSCF meneruskan ke SCSCF, SCSCF melakukan perubahan format tel URI menjadi format ENUM *request*, S-CSCF mengakses ENUM Server, ENUM Server mencari data, lalu dikembalikan kembali ke PCSCF untuk melakukan pemanggilan ulang dengan format SIP URI.
2. Dari hasil percobaan didapatkan bahwa waktu proses yang terjadi ketika melakukan panggilan dengan format SIP URI adalah 0.1071274 s (107.1 ms) , dan waktu proses yang terjadi ketika melakukan panggilan menggunakan format tel URI adalah 0.1332041 detik (133.2 ms). Sehingga dapat disimpulkan bahwa waktu proses dalam panggilan yang diperoleh ketika melewati ENUM Server menjadi lebih besar, karena adanya tambahan tahapan dalam prosesnya.
3. Dari data percobaan waktu proses dibandingkan dengan standar waktu pencarian dalam VoIP, dapat disimpulkan bahwa waktu proses baik yang tidak menggunakan ENUM maupun yang menggunakan ENUM memiliki waktu proses yang masih diterima standar maksimum yang ada, yaitu 107.1 ms dan 133.2 ms dengan standar maksimum 205-218 ms.

## DAFTAR ACUAN

- [1] AG Projects., *ENUM, NGN Routing Engine*.
- [2] Istvan Tetenyi, Gyula Szabo, Andras Kiss, Andras Toth., "ENUM in everyday practice - is it a Dream or an Opportunity." 2008, Vol. LXIII.
- [3] Charles Shen, Henning Schulzrinne., "Evaluation and Comparison of BINDS, PDNS and NAVITAS as ENUM Servers." 2006.
- [4] T.Towie, Thomas., *IMS in Next Generation Networks*. April 14, 2005.
- [5] Ericsson., "IMS-IP Multimedia Subsystem, The value of using IMS architecture." 2004.
- [6] Fraunhofer FOKUS NGNI., OpenSourceIMS. [Online] <http://www.openimscore.org/>.
- [7] Carter, Ross., Microsoft Real Time Communications: Protocols and Technologies. *Microsoft Tech Net*. [Online] July 3, 2003. [Cited: November 25, 2010.] <http://technet.microsoft.com/en-us/library/bb457036.aspx#EHAA>.
- [8] J.Rosenberg, H.Schlzrinne, G.Camarillo,A.Johnston, J.Peterson R.Sparks, M.Handley, E.Schooler., *RFC-3261-SIP : Session Initiation Protocol*. s.l. : IETF, 2002.
- [9] P.Calhoun, J.Loughney, E.Guttman, G.Zorn, J.Arkko., *RFC-3588-Diameter Base Protocol*. s.l. : IETF, 2003.
- [10] Schulzrinne, H., *RFC 3966 - The tel URI for Telephone Numbers*. December 2004.
- [11] Cisco., [Online] Cisco. [http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_5-2/enum.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_5-2/enum.html).
- [12] Saraiva, Nuno., *Importance of ENUM in IMS Network*. s.l. : Ericsson, 2009.
- [13] Telecom Malaysia., *ENUM - Operator Persective*. 2009.
- [14] Mealling, M., *RFC 2915 - The Naming Authority Pointer (NAPTR) DNS Resource Record*. September 2000.
- [15] Union, International Telecommunication., About Mobile Technolgy and IMT-2000. *International Telecommunication Union*. [Online] Desember 01, 2005. [Cited: November 20, 2010.] <http://www.itu.int/osg/spu/imt-2000/technology.html>.

- [16] Wikipedia., 3G. *Wikipedia*. [Online] [Cited: November 16, 2010.] <http://en.wikipedia.org/wiki/3G>.
- [17] —. High Speed Packet Access. *Wikipedia*. [Online] [Cited: November 15, 2010.] [http://en.wikipedia.org/wiki/High\\_Speed\\_Packet\\_Access](http://en.wikipedia.org/wiki/High_Speed_Packet_Access).
- [18] Hill.com., IP Multimedia Subsystem. [Online] [Cited: November 22, 2010.] [http://www.hill2dot0.com/wiki/index.php?title=IP\\_Multimedia\\_Subsystem](http://www.hill2dot0.com/wiki/index.php?title=IP_Multimedia_Subsystem).
- [19] Chaffin, Larry., "Chapter 8- SIP Architecture." *Building A VoIP Network with Nortel's Multimedia Communication Server 5100*. 2006.
- [20] Arslan Munir, Gordon Ross., *SIP-Based IMS Signaling Analysis for Wimax-3G Interworking Architectures*. s.l. : IEEE, 2010. 5.
- [21] Dragoz Vingarza, Peter Weik., *IMS Signaling over Current Wireless Network*. s.l. : IEEE Vehicular Technology Magazine, 2007.
- [22] Toshio Shimoe, Takamichi Sano., *IMT-2000 Network Architecture*. 2002.
- [23] Wikipedia., GPRS Core Network. *Wikipedia*. [Online] [Cited: Desember 15, 2010.] [http://en.wikipedia.org/wiki/GPRS\\_Core\\_Network](http://en.wikipedia.org/wiki/GPRS_Core_Network).
- [24] Raatikainen, Pertty., *Next Generation Network and Reliability*. 2007.