ANALISIS DAN IMPLEMENTASI METODE OFFLINE CHARGING BERBASIS SESI UNTUK LAYANAN VIDEO CALL DAN VOIP PADA JARINGAN IMS DI KOMPUTASI AWAN

ANALYSIS AND IMPLEMENTATION OF SESSION BASED OFFLINE CHARGING FOR VIDEO CALL AND VOIP OVER IMS NETWORK IN CLOUD COMPUTING

Amelia Putri Anggadhini[1], Danu Dwi Sanjoyo[2], Ratna Mayasari[3]

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University

[1]ameliagdhn@gmail.com, [2] danudwj@telkomuniversity.ac.id, [3] ratnamayasari@telkomuniversity.ac.id

Abstrak

Arsitektur IP *Multimedia Subsystem* yang menjamin konvergensi jaringan serta kualitas sesi sangat mendukung berbagai jenis konsep layanan *real-time*. Namun keunggulan IMS tersebut tidak dapat terlepas dari beberapa faktor penunjang lain, termasuk mekanisme *charging* yang handal, fleksibel dan efisien. Salah satu mekanisme *charging* yang ada di dalam arsitektur IMS adalah mekanisme *offlline charging*. Dalam penelitian ini, telah diujikan implementasi *offline charging* berbasis sesi pada jaringan IMS untuk layanan *Video Call* dan VoIP yang melibatkan Clearwater sebagai server IMS, RestComm jDiameter sebagai *Charging Data Function*, AWS sebagai penyedia komputasi awan, dan X-Lite untuk *softphone* pengguna.

Setelah dilakukan pengujian, didapatkan bahwa jaringan yang tidak mengimplementasi *charging* memiliki nilai *session setup delay* lebih rendah daripada jaringan yang mengimplementasi, dengan selisih 0,34 detik untuk layanan VoIP dan 0,25 detik untuk layanan *Video Call*. Persentase error keakuratan tarif untuk kedua layanan berada pada kisaran 1,83% hingga 3,13% atau tercatat lebih besar daripada durasi aktual. Sementara pengujian *charging* dengan parameter volume melalui pendekatan berbasis sesi telah dibuktikan tidak efektif karena pada layanan VoIP and *Video Call* secara berturut-turut terdapat Rp 0,53 dan Rp 2,80 yang hilang setiap detiknya.

Kata kunci: IMS, *Video Call*, VoIP, *session based, offline charging*

Abstract

IP Multimedia Subsystem framework that ensures network convergence as well as session quality strongly supports various types of real-time service concepts. But the benefits of IMS can not be separated from several other supporting factors, including a reliable, flexible and efficient charging mechanism. One of the charging mechanisms in IMS is the offline charging. In this research, the implementation of offline session-based charging is tested over IMS network in Video Call and VoIP using Clearwater as the IMS server, RestComm jDiameter as the Charging Data Function, AWS as the cloud provider, and X-Lite as user softphone.

After being tested, the network that doesn't implement charging has the lower session setup delay value than the network that does, with a difference of 0.34 seconds for VoIP service and 0,25 seconds for Video Call service. The error percentage of rate accuracy itself is within range of 1,83% to 3,13% or greater than the actual duration. While the volume parameter with session-based approaching is proven not effective because the VoIP and Video Call service consecutively lost Rp 0,53 and Rp 2,80 every second the services occur.

Keywords: IMS, Video Call, VoIP, Session Based, Offline Charging

1. Pendahuluan

Salah satu *framework* arsitektur berbasis IP yang berkembang adalah IP *Multimedia Subsystem* (IMS). Arsitektur IMS yang menjamin konvergensi jaringan serta kualitas sesi sangat mendukung berbagai jenis konsep layanan *real-time* yang marak saat ini. Namun, kendati jaringan seluler telah memberikan banyak pilihan teknologi dalam menyediakan akses internet, kehadiran IMS ternyata masih diperlukan untuk menjamin beberapa faktor. Salah satunya yang paling vital adalah mekanisme *charging*.

Dalam penelitian, suatu implementasi jaringan IMS sederhana yang menerapkan *offline charging* berbasis sesi dijalankan untuk menguji keakuratan pembebanan serta *delay* yang terjadi pada layanan VoIP dan *Video Call*. Server IMS yang digunakan adalah Clearwater dengan metode pengimplementasian memanfaatkan *Amazon's Elastic Compute Cloud* (EC2) yang telah disediakan oleh Metaswitch. Sementara, *Charging Data*

Function yang digunakan adalah jDiameter, sebuah CDF open-source java dari standar Diameter untuk autentikasi, autorisasi, dan akutansi yang dikembangkan oleh RestComm. Kedua komponen ini diintegrasikan untuk membangun layanan VoIP dan Video Call yang dapat diakses pengguna melalui *softphone* X-Lite.

Dari jaringan yang telah dirancang, beberapa skenario dijalankan untuk menguji perbandingan *session setup delay* dalam jaringan yang mengimplementasi *offline charging* dan yang tidak, akurasi durasi dan tarif dalam setiap panggilan, serta analisa pengujian *charging* dengan parameter volume melalui pendekatan berbasis sesi jika dihitung menggunakan *bitrate* yang disesuaikan dengan *codec* pada AVP yang diterima.

2. Dasar Teori

2.1. IP *Multimedia Subsystem*

IP *Multimedia Subsystem* (IMS) pertama kali dispesifikasi untuk jaringan seluler, terutama untuk jaringan *Universal Mobile Telecommunications System* (UMTS). IMS telah diperkenalkan dan distandarisasi oleh *Third Generation Partnership Project* (3GPP) dan *European Telecommunications Standards Institute* (ETSI) untuk menyediakan layanan telekomunikasi berbasis IP. Tujuan IMS adalah untuk menawarkan layanan seperti suara, layanan komunikasi data, pesan serta teknologi berbasis *web* dimanapun, kapanpun, dan dengan terminal apapun [1]. Arsitektur IMS terdiri dari empat lapisan yaitu *Access Layer*, *Transport Layer*, *Control Layer* dan *Application Layer* yang memiliki fungsi dan harmonisasi elemen berdasarkan protokol SIP.

2.2. Metaswitch Clearwater

Project Clearwater adalah IMS pada jaringan komputasi awan yang disponsori oleh Metaswitch Network. Cleawater mengikuti prinsip arsitektur IMS dan mendukung antar muka yang sudah distandarisasi dari sebuah jaringan *core* IMS. Tidak seperti implementasi IMS yang tradisional, Clearwater didesain untuk dijalankan pada komputasi awan. Dengan menggabungkan pola desain dan komponen perangkat lunak *open source* yang telah terbukti dalam banyak aplikasi *web* global, Clearwater mencapai kombinasi skalabilitas masif yang belum pernah terjadi sebelumnya dan efektivitas biaya yang luar biasa. Project Clearwater disponsori oleh Metaswitch Networks [2].

2.2.1. Arsitektur Clearwater

Clearwater didesain agar dapat dioptimisasi untuk implementasi pada jaringan awan. Hal ini dimaksudkan agar pola desain dapat memadai untuk penerapan dan implementasi aplikasi web dalam skala besar dengan mengadaptasi pola desain yang sesuai oleh batasan SIP dan IMS. Bagaimanapun, arsitektur Clearwater memiliki persamaan dengan arsitektur IMS walaupun tidak identikal [2].

A. Bono (*Edge Proxy*)

Node Bono membentuk SIP *edge proxy* yang memiliki skalabilitas horizontal SIP dan menyediakan antarmuka SIP IMS Gm yang sesuai serta antarmuka WebRTC ke klien. Koneksi klien memiliki beban seimbang di seluruh *node*. *Node* Bono menyediakan titik catu untuk koneksi klien ke sistem Clearwater, termasuk mendukung bermacam mekanisme NAT *traversal*.

B. Sprout (SIP *Router*)

Node Sprout berfungsi sebagai skalabilitas horizontal, menggabungkan SIP *registrar* dan otorisasi *routing proxy*, dan menangani autentikasi klien serta antarmuka ISC untuk server aplikasi. Sprout adalah tempat sebagian besar fungsi I-CSCF dan S-CSCF berada, dan sisanya disediakan oleh Dime dan didukung oleh penyimpanan data yang berumur panjang di Vellum.

C. Dime (*Diameter Gateway*)

Node Dime menjalankan *homestead* dan komponen Ralf.

D. Homestead (HSS *Cache*)

Homestead menyediakan antarmuka layanan *web* untuk Sprout atau mengambil kredensial autentikasi dan informasi profil pengguna. *Node* Homestead sendiri tidak menyimpan data, semua data pelanggan disimpan ke Vellum melalui antarmuka Cassandra Thrift.

E. Ralf (CTF)

Ralf menyediakan HTTP API yang digunakan Bono dan Sprout untuk melaporkan *billable event* yang harus disampaikan ke *Charging Data Function* melalui antarmuka *billing* Rf.

F. Vellum (*State Store*)

Seperti yang telah dijelaskan sebelumnya, Vellum digunakan untuk mempertahankan *state* jangka lama dalam penerapan. Hal ini dilakukan dengan menjalankan sejumlah kluster yang sudah diverifikasi dan dioptimalkan oleh jaringan komputasi awan.

G. Homer (XDMS)

Homer adalah XDMS standar yang digunakan untuk menyimpan dokumen pengaturan layanan MMTEL untuk setiap pengguna sistem. Dokumen dibuat, dibaca, diperbarui dan dihapus menggunakan

antarmuka standar XCAP. Seperti halnya Homestead, *node* Homer menggunakan Vellum sebagai penyimpanan data untuk semua data jangka lama.

H. Ellis

Ellis adalah portal penyediaan contoh yang menyediakan pendaftaran manual, manajemen kata sandi, *line* manajemen dan kontrol pengaturan layanan MMTEL.

2.3. *Charging*

Charging merupakan fungsi di dalam jaringan telekomunikasi dan kumpulan dari *Online Charging System* (OCS) komponen dimana informasi yang berkaitan dengan *event* berbayar dikumpulkan, diformat, ditransfer dan dievaluasi untuk memungkinkan ketentuan pemakaian dari pengguna mana yang akan dikenakan biaya. Rf dan Ro adalah poin referensi dari *Charging Trigger Function* (CTF) ke *Charging Data Function* (CDF) dan *Online Charging Function* (OCF), dan dimaksudkan untuk mentransport *charging events* [3].

Proses *charging* pada jaringan 3GPP sendiri memiliki dua mekanisme yang dapat dibedakan menjadi *offline charging* dan *online charging*. Kunci utama dari perbedaan dua mekanisme ini terletak pada poin referensi yang digunakan.

Pada *online charging* poin referensi yang digunakan adalah Ro, dimana informasi *charging* dapat berpengaruh pada saat suatu sesi layanan sedang berlangsung. Sementara *offline charging* menggunakan poin referensi Rf, dimana hasil akhir dari mekanisme *charging* ini adalah meneruskan CDR ke *Billing Domain* tanpa mempengaruhi *credit* pengguna secara *real-time*.

2.3.1. *Offline Charging*

Dalam *offline charging* terdapat terdapat tiga fungsi *charging* yang terlibat, yaitu *Charging Trigger Function* (CTF), *Charging Data Function* (CDF), dan *Charging Gateway Function* (CGF) [3].

A. *Charging Trigger Function* (CTF)

CTF membangkitkan *charging event* berdasarkan observasi penggunaan sumber daya jaringan. Di setiap jaringan dan servis layanan yang menyediakan informasi *charging*, CTF adalah poin utama untuk mengumpulkan informasi yang berkaitan dengan *chargable event* dalam elemen jaringan, menghimpun informasi ini ke dalam *charging event* yang sesuai, kemudian mengirimnya ke *Charging Data Function*. CTF adalah komponen wajib yang harus ada di elemen jaringan yang menyediakan fungsionalitas *offline charging*.

B. *Charging Data Function* (CDF)

CDF menerima *charging event* dari CDF melalui poin referensi Rf. CDF kemudian menggunakan informasi yang ada di *charging event* untuk membangun *Charging Data Record* (CDR). Hasil dari proses di dalam CDF adalah CDR dengan konten dan format yang baik.

C. *Charging Gateway Function* (CGF)

CDR yang diproduksi CDF kemudian dilanjutkan ke CGF melalui poin antarmuka Ga sebelum diteruskan ke *Billing Domain*. Hubungan antara CDF dan CGF adalah banyak CDF dapat memberi banyak input CDR dalam satu CGF.

2.3.2. *Session Based Charging*

Charging berbasis sesi dimulai ketika sebuah panggilan dideteksi oleh perangkat jaringan yang menanganani panggilan tersebut. *Chargeable event* ini kemudian dipetakan menjadi *charging event* yang telah dispesifikasi pada tingkat tengah spesifikasi teknikal *charging* yang diaplikasikan pada perangkat jaringan.

Seperti yang sudah disebutkan sebelumnya, inisiasi *offline charging* ditransfer ke CDF melalui poin referensi Rf. Pada terminasi dari sesi pelanggan atau ketika terdapat *chargeable event* yang terjadi, *charging event* selanjutnya akan dikirim untuk sesi dari perangkat jaringan ke CDF. Setelah hal tersebut selesai, CDR baru akan dibuka oleh CDF untuk sesi yang sama. Terakhir, CDR akan ditransfer dalam bentuk CDR *file*, bersama dengan CDR yang lain yang memiliki tipe sama maupun berbeda sesuai dengan konfigurasi transfer *file* yang ditetapkan oleh operator [4].

2.4. RestComm jDiameter

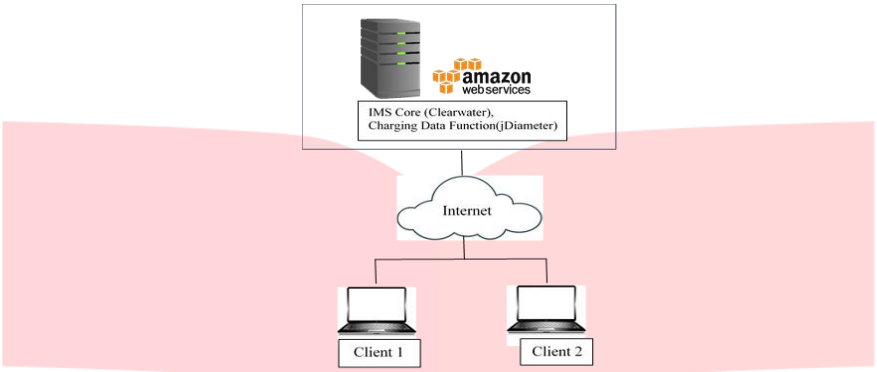
RestComm jDiameter merupakan salah satu *open source* Java dari standar Diameter. Dengan menerapkan *Base Protocol* serta beberapa aplikasi yang penting dan banyak digunakan, RestComm Diameter memungkinkan pengembangan komponen IMS yang cepat, seperti *Application Server* (AS), *Home Subscriber Server* (HSS), *Call Session Control Function* (CSCF), *Subscriber Location Function* (SLF), dan lain-lain.

Aplikasi yang didukung oleh jDiameter termasuk *Base*, *Credit-Control Application*, Ro, Rf, Sh, Gx, Cx/Dx, Gq, S6a dan banyak lainnya. jDiameter juga memungkinkan fitur arsitektur yang dapat diperluas dan memungkinkan tambahan aplikasi Diameter [9].

3. Hasil Implementasi dan Pembahasan

3.1. Topologi Jaringan

Penelitian ini menggunakan topologi jaringan seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Topologi Jaringan

Komponen-komponen yang menyusun topologi ini antara lain;

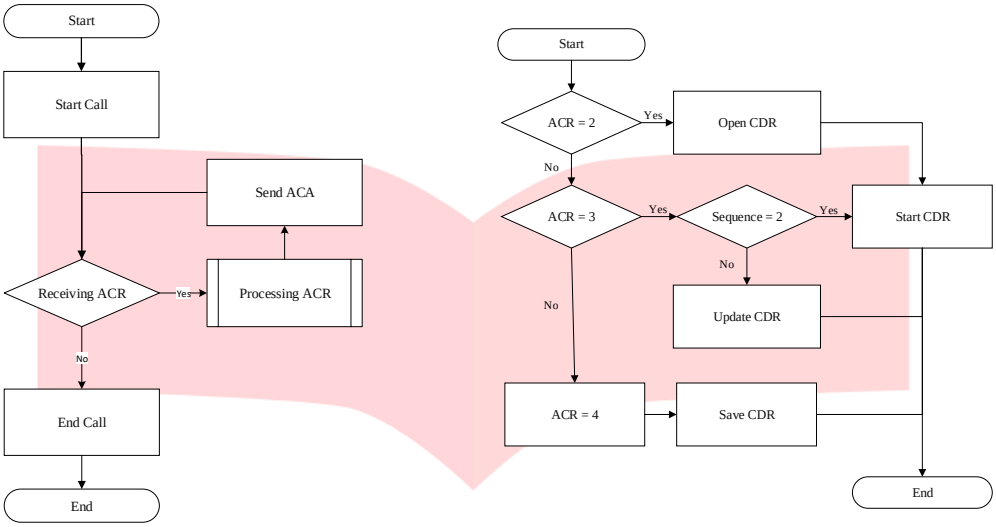
a. Clearwater versi Nazgûl sebagai jaringan IMS *core*
   Server IMS ini disediakan oleh Metaswitch Project dan telah dilengkapi dengan komponen IMS semacam P-CSCF, I-CSCF, S-CSCF, Diameter *Gateway* dan BGCF.
b. RestComm jDiameter versi 1.7.0 sebagai *Charging Data Function*
   Sebuah mekanisme implementasi *offline charging* yang telah disesuaikan oleh standar protokol Diameter untuk autentikasi, autorisasi, dan akutansi. Aplikasi ini meliputi Base, aplikasi *Credit-Control*, *Accounting Request* juga antarmuka Rf untuk *offline charging*.
c. X-Lite versi 5.2.0 sebagai *softphone* pengguna
   Pada bagian *end user*, *softphone* X-Lite yang telah dikonfigurasi berdasarkan identitas yang telah registrasi pada server IMS. Dari fitur statistik panggilan yang disediakan oleh X-Lite pula, akan dianalisa paket yang diterima pengguna selama terjadi panggilan.
d. *Amazon Web Service's* EC2 t2.*small* sebagai instansi komputasi awan
   EC2 digunakan sebagai instansi yang menjalankan server IMS dalam lingkungan komputasi awan. EC2 menyediakan akses komputasi awan kepada perusahaan dan individu untuk menyewa komputer *storage* yang bisa digunakan sebagai pengembangan aplikasi secara *online*. Clearwater merupakan salah satu pengembang yang memanfaatkan layanan ini.
e. Wireshark versi 2.6.1 sebagai *Network Analyzer*
   Wireshark digunakan sebagai alat untuk mencatat, menangkap dan menganalisa *delay* yang terjadi pada mekanisme *offline charging* untuk kemudian dibandingkan dengan jaringan yang tidak mengimplementasi *charging*.
   Dalam penerapan mekanisme *offlline charging* terjadi interaksi yang berlangsung antara CTF dengan CDF ketika suatu panggilan terdeteksi. Interaksi tersebut berisi pesan *Accounting Request* (ACR) dan *Accounting Answer* (ACA) yang ditukar melalui poin referensi Rf. Pertukaran pesan ini kemudian digunakan sebagai acuan dalam menentukan informasi yang dibutuhkan oleh *Charging Data Record* untuk membuat tagihan kepada pengguna.

3.2. Mekanisme *Offline Charging*

Ketika suatu panggilan dideteksi oleh server, CTF akan mengirimkan ACR kepada CDF. CDF yang menerima pesan ACR dari CTF kemudian akan melanjutkan proses tersebut pada sub-proses *"Processing* ACR". Sementara jika CDF tidak menerima pesan ACR dari CTF maka panggilan akan dinyatakan berakhir. Pada sub-proses "*Processing ACR*" terdapat beberapa skenario yang mungkin terjadi. Parameter yang menjadi pembeda dalam skenario ini adalah *Accounting Record Number* dan *Account Record Type* AVP yang didetekesi oleh CDF.

Saat CDF menerima pesan *Account Record Type* dengan kode 2 (START), maka CDF akan membuka CDR baru untuk sesi yang baru saja berlangsung. Namun jika kode yang diterima adalah 3 (INTERIM), maka asumsi yang terjadi adalah panggilan masih berlangsung. Jika demikian, akan dicek urutan *Accounting Record Number* atau *sequence* yang ditangkap. Jika ACR sudah dikirimkan sebanyak dua kali, maka CDR bisa langsung dibuat.

Sedangkan jika kode 4 (STOP) yang diterima, hal ini mengindikasikan bahwa panggilan telah diterminasi. CDR terkait panggilan tersebut kemudian akan disimpan.



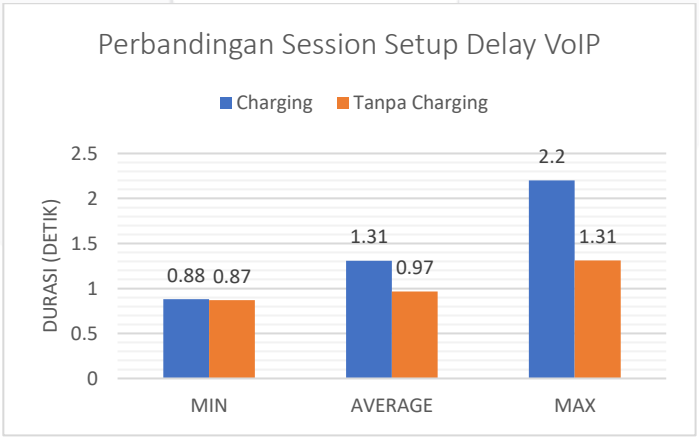
Gambar 3. 2 Diagram Alir *Offline Charging* Berbasis Sesi dan Subproses ACR

4. Pengukuran dan Analisis

4.1. Pengujian *Session Setup Delay*

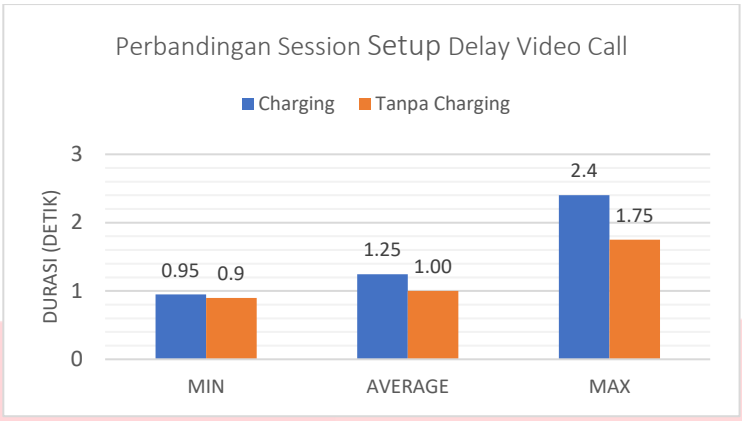
Pengujian pertama yang dijalankan adalah membandingkan *session setup delay* yang terjadi pada jaringan yang mengimplementasi sistem *charging* dan jaringan yang tidak mengimplementasi. *Session setup delay* sendiri merupakan periode yang dimulai ketika penelepon menekan digit terakhir dan pesan INVITE dikirim hingga status 180 (*Ringing*) diterima. Berdasarkan standar ITU-T E0771, nilai waktu *session setup delay* yang dianjurkan adalah kurang dari tiga detik [10].

Dari hasil pengujian didapatkan bahwa layanan VoIP yang menerapkan mekanisme *offline charging* memiliki nilai *delay* rata-rata 1,97 detik, dengan waktu minimum sebesar 0,88 detik dan waktu maksimum sebesar 2,2 detik. Sementara jaringan yang tidak mengimplementasi memiliki nilai *delay* rata-rata 1,31 detik, dengan waktu minimum sebesar 0,87 detik dan waktu maksimum sebesar 1,31 detik.



Gambar 4. 1 Grafik Perbandingan *Session Setup Delay* pada VoIP

Pada layanan *Video Call*, jaringan yang mengimplementasi *charging* memiliki nilai *delay* rata-rata 1,25 detik, dengan waktu minimum sebesar 0,95 detik dan waktu maksimum sebesar 2,4 detik. Sedangkan jaringan yang tidak mengimplementasi *charging* memiliki nilai *delay* rata-rata 1,00 detik, dengan waktu minimum sebesar 0,90 dan waktu maksimum sebesar 1,75 detik.



Gambar 4. 2 Grafik Perbandingan *Session Setup Delay* pada *Video Call*

Jika ditinjau dari dua hasil pengujian, dapat disimpulkan bahwa jaringan yang tidak mengimplementasi *charging* memiliki nilai *delay* lebih rendah daripada jaringan yang mengimplementasi. Selisih *delay* secara berturut-turut pada layanan VoIP adalah 0,34 detik dan pada layanan *Video Call* sebesar 0,25 detik.

Perbedaan *delay* tersebut terjadi karena dalam jaringan yang mengimplementasi *charging* terdapat komunikasi tambahan yang dibutuhkan dari Ralf pada Clearwater yang berperan sebagai CTF ke jDiameter yang berperan sebagai CDF. Komunikasi tambahan tersebut adalah *message forwarding* berupa ACR START dari CTF yang mengindikasikan kepada CDF bahwa terdapat *billable event* yang akan berlangsung. Ketika menerima pesan ini, CDF kemudian akan merespon dengan mengirimkan ACA ke CTF sebagai persetujuan bahwa sesi yang bersangkutan dapat dirilis.

Adapun dari segi kelayakan, baik jaringan yang menerapkan dan tidak menerapkan mekanisme *charging* masih memenuhi standar yang ditetapkan oleh ITU-T untuk nilai *session setup delay* yaitu 1.97 detik, 1,31 detik, 1,25 detik dan 1,00 detik atau kurang dari tiga detik.

4.2. Pengujian Keakuratan Durasi dan Tarif

Dalam mekanisme *charging* salah satu hal yang sangat perlu diperhatikan adalah kemampuan CDR untuk menghasilkan rekam pembebanan tarif yang sesuai dengan durasi panggilan aktual dari para pengguna. Pada pengujian ini, dianalisa bagaimana keakuratan tarif dan pencatatan durasi yang dihasilkan oleh CDR untuk setiap panggilan pada layanan *Video Call* dan VoIP.

Memanfaatkan fitur *Wireshark*, penentuan tarif dasar yang digunakan dalam pengujian ini diambil dari penghitungan jumlah rata-rata *bytes* yang ditransmisikan dalam suatu panggilan. Kemudian jumlah tersebut dikalikan dengan tarif dasar operator yaitu Rp 1,00/Kb [7] sebelum dikonversi menjadi satuan per sepuluh detik.

Proses ini dapat dijabarkan lebih lanjut dalam persamaan berikut;

$$Tarif = \frac{\text{Jumlah bytes}}{1024 \text{ x durasi}} \; Rp \; 1{,}00 \text{ x } 10 \; detik \qquad (1)$$

Dengan menggunakan persamaan (2), kemudian dihitung nilai presentase error untuk rata-rata *delay* masing-masing skenario.

$$Presentase \; Eror = \frac{\text{Delay durasi}}{\text{Durasi Aktual}} \text{ x } 100\ \% \qquad (2)$$

Tabel 4. 1 Tabel Presentase Error

Tipe	Delay Rata-Rata	Presentase Error
VoIP 60 Detik	1.1	1.83%
VoIP 90 Detik	1.66	2.77%
VoIP 120 Detik	1.414	2.36%
Video Call 60 Detik	1.374	2.29%
Video Call 90 Detik	1.788	2.98%
Video Call 120 Detik	1.876	3.13%

Jika dilihat dari Tabel 4.1, seluruh persentase error bernilai pada kisaran satu persen. Hal tersebut menunjukkan bahwa pada setiap panggilan terdapat *delay* sekitar satu detik yang terjadi sebelum server benar-benar melakukan terminasi.

Dalam teorinya, nilai persentase error *charging* yang terbaik adalah 0%. Nilai tersebut membuktikan bahwa mekanisme *charging* yang dijalankan telah berhasil mengkonversi antara durasi aktual dari suatu panggilan dengan tarif yang harus mereka bayarkan. Namun pada kenyataannya, terdapat banyak faktor yang dapat menyebabkan durasi paa CDR melebihi nilai durasi aktual. Pada lingkungan komputasi awan, salah satu faktor tersebut adalah latensi di lingkungan komputasi awan yang tidak dapat diprediksi karena pusat data layanan komputasi secara fisik tidak berada pada lingkungan geografis yang sama.

Latensi ini kemudian akan mempengaruhi proses pengiriman ACR dengan tipe STOP yang menjadi penyebab adanya perbedaan durasi aktual dengan durasi yang tercatat pada CDR. Untuk melihat tingkat akurasi durasi, dari beberapa hasil percobaan yang didapat telah diambil rata-rata masing-masing layanan agar dapat dianalisa tingkat presentase *error*-nya.Proses pengiriman ACR dengan tipe STOP ini merupakan salah satu penyebab adanya perbedaan durasi aktual dengan durasi yang tercatat pada CDR. Namun selain faktor tersebut, terdapat faktor lain juga yang mempengaruhi *delay* seperti terlambatnya deteksi SIP BYE pada server dan kesalahan *timer* pada *softphone* yang digunakan.

4.3. Pengujian Tarif dengan Parameter Volume

Dalam jDiameter, terdapat beberapa jenis AVP yang bisa digunakan untuk mempermudah kustomisasi *charging*. Oleh karena itu, dengan memanfaatkan salah satu AVP yang dimiliki jDiameter akan dibuktikan apakah *charging* berbasis volume dengan pendekatan sesi dapat diwujudkan. Metode yang digunakan adalah dengan cara mendeteksi jumlah AVP SDP *Media Component* pada *Session Description Protocol* ketika pesan ACR START diterima. Untuk layanan VoIP, jumlah AVP *SDP Media Component* diterima adalah dua, sementara untuk layanan *Video Call* jumlah AVP SDP *Media Component* yang diterima adalah empat.

Jumlah AVP ini kemudian akan dibuat sebagai referensi dalam mendeteksi layanan mana yang sedang digunakan oleh pengguna. Dengan fungsi *switch*, algoritma pada CDF akan menyesuaikan pentarifan sesuai dengan deteksi layanan yang diterima.

Penghitungan tarif untuk VoIP ditentukan dengan menggunakan persamaan (3), Dimana 13,2 Kbit/s merupakan *bitrate* dari *codec* audio yang tersedia yaitu GSM dan Rp 1,00 adalah tarif dasar operator untuk setiap KiloByte data.

$$Tarif\ VoIP\ = \frac{13{,}2\ \text{Kbit} * \text{Durasi}}{8}\ Rp\ 1{,}00 \qquad (2)$$

Sementara penghitungan tarif untuk *Video Call* ditentukan dengan menggunakan persamaan (4), dimana 13,2 Kbit/s merupakan *bitrate* GSM untuk audio, 90 Kbit/s merupakan *bitrate* dari *codec* video yang tersedia yaitu H.263, dan Rp 1,00 adalah tarif dasar operator untuk setiap KiloByte data. *Bitrate codec* GSM dan H.263 dijumlahkan karena layanan *Video Call* membutuhkan komunikasi audio dan video dalam waktu bersamaan.

$$Tarif\ Video\ Call = \frac{(13{,}2\ \text{Kbit} + 90\ \text{Kbit}) * \text{Durasi}}{8}\ Rp\ 1{,}00 \qquad (3)$$

Setelah tarif ditentukan, maka selanjutnya akan dibuktikan apakah tarif berdasarkan *bitrate* ini sesuai dengan volume pemakaian aktual. Pengukuran jumlah *bytes* yang ditransmisikan selama panggilan dilakukan dengan memanfaatkan fitur *Wireshark.*

Tabel 4. 2 Hasil Pengujian VoIP dan *Video Call*

Tipe	Selisih Volume (Kb)
VoIP 60 Detik	31.59
VoIP 90 Detik	49.00
VoIP 120 Detik	63.48
Rata-rata per detik	0.53

Tipe	Selisih Volume (Kb)
Video Call 60 Detik	182.06
Video Call 90 Detik	257.00
Video Call 120 Detik	316.52
Rata-rata per detik	2.80

Selisih volume pada Tabel 4.2. merupakan selisih dalam satuan KiloByte. Maka dari itu dapat disimpulkan bahwa selisih volume juga merupakan selisih tarif yang seharusnya dibayarkan oleh pengguna. Dapat dilihat bahwa pada layanan VoIP and *Video Call* berturut-turut terdapat Rp 0,53 dan Rp 2,80 yang hilang setiap detiknya.

Adanya selisih ini disebabkan oleh CDF yang menerapkan *bitrate* tetap sebagai faktor pengali. Sementara *bitrate* yang digunakan pada pengujian adalah *bitrate* minimum yang dibutuhkan agar komunikasi VoIP maupun *Video Call* bisa terjadi. Namun pada prakteknya, *bitrate* seharusnya bersifat dinamis atau dapat berubah sesuai dengan *bandwidth* jaringan. Seperti pada pengujian ini, *bitrate* pada layanan VoIP tercatat maksimum mencapai 18 kBits/detik, sedangkan *bitrate* yang ditetapkan pada CDF adalah 13,2 kBits/detik.

Oleh karena itu dapat ditarik kesimpulan bahwa *charging* berbasis volume dengan pendekatan sesi tidak efektif. Mengingat protokol RTP yang digunakan dalam layanan VoIP dan *Voice Call* merupakan protokol yang didesain untuk *end-to-end*, server tidak mampu menangkap besar trafik yang ditransmisikan ketika panggilan berlangsung. Sementara, untuk mendapatkan hasil akurat dengan pendekatan ini, server dituntut untuk mampu mendeteksi besar *bitrate* di setiap panggilan yang kemudian dapat ditetapkan sebagai faktor pengali di CDF.

5. Kesimpulan dan Saran

5.1. Kesimpulan

1. Adanya *message forwarding* untuk pengiriman ACR START dari Ralf kepada jDiameter membuat jaringan yang mengimplementasi *charging* memiliki nilai *session setup delay* lebih rendah daripada jaringan yang tidak mengimplementasi, dengan selisih rata-rata 0,34 detik untuk layanan VoIP dan 0,25 detik untuk layanan *Video Call.*
2. Nilai *session setup delay* pada jaringan yang mengimplementasi dan tidak mengimplementasi *charging* masih memenuhi standar yang ditetapkan oleh ITU-T untuk nilai *session setup delay* yaitu 1.97 detik, 1,31 detik, 1,25 detik dan 1,00 detik atau kurang dari tiga detik.
3. Persentase *error* keakuratan tarif untuk layanan VoIP dan *Video Call* berada pada kisaran 1,83% hingga 3,17% atau tercatat lebih besar pada kisaran satu detik daripada durasi aktual. Terjadinya *delay* saat terminasi panggilan yang disebabkan oleh pengiriman ACR STOP dari Ralf kepada jDiameter adalah faktor yang mempengaruhi perbedaan durasi aktual dengan data durasi pada CDR.
4. Pengujian *charging* dengan parameter volume melalui pendekatan berbasis sesi tidak efektif karena pada layanan VoIP and *Video Call* secara berturut-turut terdapat 0,53 Kb dan 2,34 Kb data yang hilang setiap detiknya. Protokol RTP yang didesain untuk *end-to-end* tidak dapat menangkap besar trafik yang ditransmisikan ketika panggilan berlangsung maupun *bitrate* dari yang bisa berubah-ubah sesuai dengan *bandwidth* jaringan pengguna.

5.2. Saran

1. Perlu dilakukan percobaan infrastruktur *cluster* Clearwater untuk melihat performansi jika *node* masing-masing komponen dijalankan dalam mesin yang berbeda.
2. Dibutuhkan eksperimen dengan interim interval yang berbeda agar didapatkan nilai interim yang optimal guna meminimalisasi persentase error.
3. Diperlukan penambahan komponen jaringan seperti *Session Border Controller* (SBC) atau modul tambahan PCEF dan PCRF untuk bisa mendapatkan volume trafik yang akurat pada protokol RTP.

6. Daftar Pustaka

[1] S. Kamel, M. Kamel, A. Tag Eldien, E. Dein, N. H Hegazi, H. M Abd, E. Kader and H. Harb, "A Proposed IP Multimedia Subsystem Security Framework for Long Term Evaluation (LTE) Networks.," *International Journal of Computer Science and Information Technologies,* vol. 6, no. 4, pp. 3992-3998, 2015.

[2] Clearwater, "About Clearwater," Metaswitch, 1 January 2015. [Online]. Available: http://www.projectclearwater.org. [Accessed 5 12 20117].

[3] R. Kuhrie, G. Gormer, M. Schlager and G. Carle, "Charging in the IP Multimedia Subsystem: A Tutorial," *Communications Magazine, IEEE,* vol. 45, pp. 92-99, 2007.

[4] 3rd Generation Partnership Project, "Diameter charging applications (Release 9)," *Technical Specification Group Services and System Aspects,* 2011.

[5] RestComm, "jDiameter," 5 12 2017. [Online]. Available: http://documentation.telestax.com. [Accessed 5 12 2017].

[6] N. Alamsyah, "Analisis dan Implementasi Metode Offline Charging Session Time Based pada layanan IPTV(VoD) dalam jaringan IMS," Telkom University, Bandung, 2013.

[7] H. Triana, "Memahami Tarif Hitung Pulsa dan Quota Data Penggunaan Media Sosial," Rabu April 2017. [Online]. Available: http://www.industry.co.id. [Diakses Kamis Juli 2018].