

## IMPLEMENTASI DAN ANALISIS METODE *FAILOVER* PADA SISTEM *REDUNDANT DEDICATED SERVER* DAN *CLOUD SERVER* UNTUK LAYANAN VOIP

### *IMPLEMENTATION AND ANALYSIS OF FAILOVER METHOD ON REDUNDANT SYSTEM DEDICATED SERVER AND CLOUD SERVER FOR VOIP SERVICE*

Devyta Asterina<sup>1</sup>, Dr. Ir. Rendy Munadi, M.T.<sup>2</sup>, Ratna Mayasari, S.T., M.T.<sup>3</sup>  
<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom  
<sup>1</sup>[devytasterina@students.telkomuniversity.ac.id](mailto:devytasterina@students.telkomuniversity.ac.id), <sup>2</sup>[rendymunadi@telkomuniversity.ac.id](mailto:rendymunadi@telkomuniversity.ac.id),  
<sup>3</sup>[ratnamayasari@telkomuniversity.ac.id](mailto:ratnamayasari@telkomuniversity.ac.id)

#### Abstrak

Layanan VoIP (*Voice Over IP*) yang dapat melewati trafik suara, data, dan video dewasa ini semakin berkembang di masyarakat. Kebutuhan akan layanan yang bersifat *realtime* berbasis IP ini pun menuntut performansi yang baik pula. Salah satu yang memegang peranan penting adalah server yang berperan sebagai penyedia layanan. Untuk itu perlu memiliki suatu cara agar tetap menjaga performansi layanan VoIP yang memenuhi standar “baik” dari segi QoS adalah dengan menggunakan *redundant* server apabila *master* server (utama) mengalami *down*. Saat ini, suatu infrastruktur belum sepenuhnya bermigrasi menggunakan *cloud computing*. *Dedicated* server masih digunakan mengingat akan performansi yang dihasilkan lebih baik dibanding dengan *cloud* server. Peranan *backup* server pun diperlukan apabila terjadi gangguan dan menggantikan kinerja *master* server.

Dalam tugas akhir ini diimplementasikan mengenai *redundant* server VoIP antara *dedicated* server dan *cloud* server. Dimana *dedicated* server berperan sebagai *master* server, dan *cloud* server berperan sebagai *backup* server yang menggantikan kinerja *dedicated* server saat mengalami *down*. Hal ini untuk mewujudkan kondisi *high availability* dengan menghasilkan *downtime* yang kecil baik saat kondisi yang terencana (*planned*) ataupun saat kondisi (*unplanned*) sehingga layanan VoIP akan tetap dapat digunakan oleh *client*.

Dengan sistem yang diimplementasikan diatas, hasil untuk segi *downtime* yang ditangani oleh *backup* server pada *cloud* adalah 0.8537 detik/tahun untuk kondisi *planned* dan bernilai 1.1858 detik/tahun untuk kondisi *unplanned*, nilai ini lebih kecil dibandingkan dengan tanpa menggunakan server *backup* yang memiliki *downtime* 172.6 menit/tahun. Untuk nilai *availability* yang dimiliki berdasarkan hasil perhitungan adalah 99.9999% baik untuk kondisi *planned* maupun *unplanned* yang telah memenuhi rekomendasi “*The-Six Nines*”. Sedangkan untuk *throughput* yakni untuk mengetahui kondisi server sesaat sebelum *failover* adalah 10790.85367 Bps dan setelah *failover* yakni pada kondisi *planned* adalah 10720.54967 Bps serta kondisi *unplanned* adalah 10718.05172 Bps yang berarti tidak mengalami perubahan yang signifikan meskipun telah terjadi *failover*. Dengan ini implementasi ini layak untuk diimplementasikan.

**Kata kunci:** *Failover, Downtime, Master server dedicated, Backup server cloud, High availability, VoIP*

#### Abstract

VoIP services (*Voice Over IP*) that can pass voice traffic, data, and video is growing in today's society. The need of *realtime* services which are based on IP is also demanding a good performance. One of the important role is a server that acts as a service provider. It is necessary to have a way in order to keep the performance of VoIP services meet the better quality in terms of QoS by using *redundant* server when the *master* server (*primary*) experiences *down*. Currently, an infrastructure is not yet fully migrated to use *cloud computing*. *Dedicated* servers still in use considering their performance is better than the *cloud* server. The role of *backup* server is required in the event of disruption and will replace the *master* server performance.

In this final task implemented *redundant* server on the VoIP services between *dedicated* server and *cloud* server. Where *dedicated* server as the *master* server and *cloud* server as a *backup* server that replace the current performance of a *dedicated* server when it experiences *down*. It is to achieve *high availability* condition and produces little *downtime* when the condition is *planned* or the condition is *unplanned* so that VoIP services will continue to be used by the *client*.

With the implemented system above, the results in terms of *downtime* handled by the *backup* server in *cloud* server is 0.8537 seconds/year for *planned* condition and is worth 1.1858 seconds/year for *unplanned* condition, this value is smaller than not using a *backup* server which has the *downtime* 172.6 minutes/year. For *availability* value owned by the result of the calculation is 99.9999% both for *planned* and *unplanned* condition that have met the recommendation of “*The Six Nines*”. *Throughput*, determine the condition of server just before the *failover* is 10790.85367 Bps and after the *failover* is 10720.54967 Bps for *planned* condition and 10718.05172 Bps for *unplanned* condition which means it has not a significant change even though there has been a *failover*. Therefore, the implementation is feasible to implemented.

**Keyword:** *Failover, Downtime, Master server dedicated, Backup server cloud, High availability, VoIP*

## 1. Pendahuluan

Dewasa ini layanan VoIP (*Voice Over IP*) sudah tidak asing lagi untuk diterapkan sebagai layanan yang mampu melewati trafik suara, data, dan video di jaringan IP. Paket yang dikirimkan pun bersifat *realtime*, maka dituntut memiliki performansi yang baik pula. Performansi suatu layanan tersebut tergantung oleh *server* yang menanganinya. Salah satu server yang bersifat *dedicated* masih diperlukan guna menunjang kebutuhan performansi yang baik dan stabil. Selain itu *Cloud* server hadir dengan teknologi virtualisasinya, yakni dengan konsep pembagian *single hardware* menjadi beberapa *virtual resource*. Namun, saat ini banyak perusahaan yang belum sepenuhnya melakukan migrasi ke arah *cloud* server. Salah satu faktornya adalah *hardware* yang masih belum dapat dimiliki dikarenakan *cost* yang relatif mahal dan perlunya ada proses migrasi server terkait dari *dedicated* menjadi *virtual* yang tentu saja akan mengganggu performansi yang dihasilkan, terlebih jika hal tersebut menyangkut kebutuhan yang sangat vital bagi perusahaan. Masalah mulai timbul apabila *dedicated* server bertindak sebagai *master server* mengalami *down* atau gagal server. Maka diperlukan sistem redundansi untuk melakukan *failover* dari *master server* menuju *backup server*. *Cloud* server dirasa pantas untuk berperan sebagai *backup server* karena cenderung fleksibel dari segi monitoring dan *hardware*. Oleh karena itu diimplementasikannya *redundant* server antara *dedicated* server dan *cloud* server diharapkan dapat mewujudkan kondisi *high availability* sehingga kebutuhan *client* akan layanan VoIP tetap terjaga.

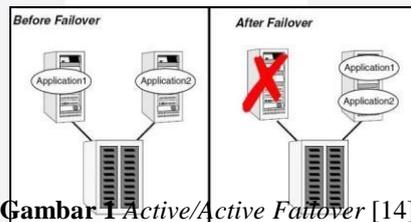
## 2. Dasar Teori

### 2.1 Failover

Failover merupakan suatu proses pengalihan kinerja ke endpoint cadangan (*redundant endpoint*) yang telah tersinkronisasi dengan *endpoint* aktif sebelumnya. Kondisi ini dapat terjadi karena disebabkan oleh kegagalan pada endpoint aktif saat melakukan kinerja sebagai penyedia layanan. [4]

#### 2.1.1. Active/Active Failover

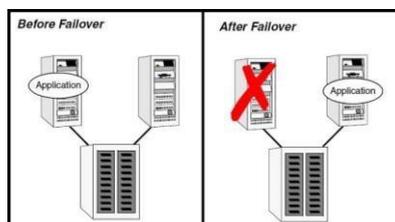
Pada *active/active failover*, seluruh *node* secara aktif menjalankan masing-masing aplikasi yang berbeda yang merupakan beban dari masing-masing *node*. Apabila pada salah satu *node* mengalami kegagalan, *server* lain akan mengambil alih beban dan akan menjalankan seluruh aplikasi yang ada. [4]



Gambar 1 Active/Active Failover [14]

#### 2.1.2. Active/Passive Failover

Pada *active/passive failover*, salah satu *node* menjalankan aplikasi yang merupakan beban kerjanya, dan *node* lain berstatus *stand by* dengan tidak menjalankan beban kerja apapun. Pada saat terjadi kegagalan di *node* utama, *node* lain akan siap bertindak sebagai *backup* yang selanjutnya akan menjalankan aplikasi seperti pada *node* utama. [6]



Gambar 2 Active/Passive Failover [15]

## 2.2 Dedicated Server

*Dedicated* server merupakan server yang tidak terbagi dengan siapapun, artinya pihak terkait memiliki kendali penuh atau utuh terhadap server tersebut. Hal ini menjadikan *dedicated* server menjadi lebih aman, dan dari segi performansi pun memiliki kualitas yang baik, mengingat harga yang ditawarkan pun relatif mahal.

Penggunaan dari *dedicated* server sendiri biasanya untuk skala besar, karena untuk permintaan layanan yang tinggi pula. Namun, ada beberapa kekurangan yang dimiliki oleh *dedicated* server antara lain karena spesifikasi yang dimilikinya bersifat tetap, artinya kapasitas penyimpanan yang dimilikinya pun terbatas dan juga rentan terhadap kegagalan hardware. Selain itu pengguna harus tetap membayar *power*/daya secara maksimum meskipun sedang tidak melakukan proses apapun. [5]

**2.3 Cloud Server**

Cloud server seperti halnya virtualisasi server, hal ini membuat pengguna dapat mengatur sendiri sumber daya hardware yang mereka butuhkan (tergantung jenis layanan yang digunakan) dan segala proses penginstallan hardware beserta penyediaannya akan diatur oleh vendor terkait.

Sistem yang digunakan biasanya *pay-for-what-you-use*, karena dalam hal ini pengguna dapat mengatur sendiri sumber daya nya sesuai dengan kebutuhan dan cukup membayar *power*/daya sesuai yang telah diatur tergantung dengan kebutuhannya pula. [5]

**2.4 Voice Over Internet Protocol (VoIP)**

Voice Over IP merupakan teknologi pengiriman *voice* (dimungkinkan juga untuk tipe data multimedia yang lain) secara *real time* antara dua atau lebih *user*/partisipan dengan melewati jaringan yang menggunakan protokol-protokol internet, dan melakukan pertukaran informasi yang dibutuhkan untuk mengontrol pengiriman *voice* tersebut.[14]

**2.5 Quality of Service**

*Quality of Service* merupakan kemampuan suatu jaringan untuk menyediakan tingkat jaminan layanan yang berbeda-beda, sesuai dengan platform teknologi yang digunakan. Kualitas paket data pada jaringan VoIP, ditunjukkan dengan parameter-parameter *Quality of Service* (QoS) salah satunya adalah *throughput*. [13]

**2.5.1 Throughput**

*Throughput* dalam jaringan telekomunikasi merupakan rata-rata pengiriman sukses dalam suatu pengiriman (satuan bps). Sedangkan, sistem *throughput* atau jumlah *throughput* merupakan jumlah rata-rata paket data yang sukses dikirimkan oleh semua terminal pada sebuah jaringan. Pada umumnya *throughput* maksimum sering dikenal sebagai *throughput*. *Throughput* maksimum dari sebuah titik atau jaringan komunikasi menandakan kapasitas dari jaringannya. Secara matematis *throughput* dapat ditunjukkan dengan persamaan berikut ini: [13]

$$\frac{\text{Jumlah data yang sukses diterima}}{\text{Jumlah total waktu pengiriman paket}}$$

**Persamaan 1** Perhitungan *Throughput* [13]

**2.6 High Availability**

Berdasarkan dokumen ISO 2382-14, 1997 availability dapat didefinisikan sebagai: “Kemampuan sebuah alat untuk berada dalam kondisi siap pakai sesuai fungsi yang diinginkan pada waktu tertentu atau kapanpun dalam interval waktu tertentu, diasumsikan bahwa sumber eksternalnya, bila diperlukan, adalah tersedia.” Secara garis besar availability merupakan nilai persentase jumlah waktu suatu jaringan mampu memberikan layanan dibandingkan dengan jumlah waktu yang diharapkan. Sedangkan rata-rata waktu suatu sistem atau jaringan dalam kondisi down atau tidak mampu memberikan layanan disebut downtime. Sedangkan uptime didefinisikan sebagai waktu rata-rata suatu sistem atau jaringan dalam keadaan operasional.[1] Nilai availability dapat dihitung menggunakan persamaan [9] sebagai berikut.

$$\frac{\text{Waktu operasional}}{\text{Waktu total}} = 100\%$$

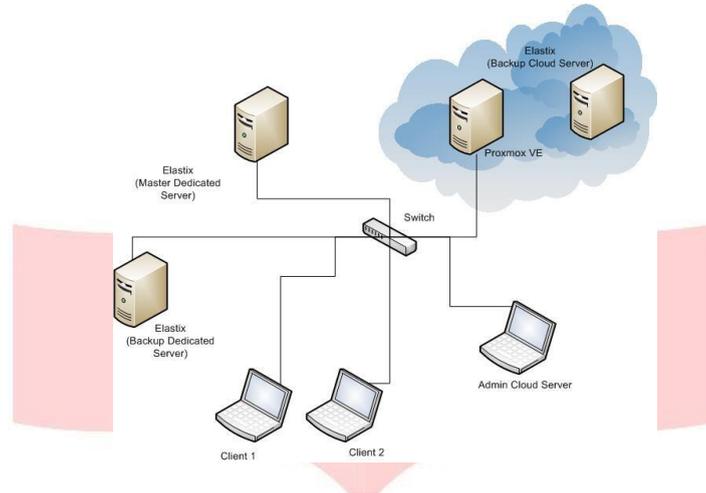
**Persamaan 2** Perhitungan *Availability* [9]

Name	Percentage of Uptime per Year	Effective Downtime per year
One nine	90%	36 days, 12 hours
Two nines	99%	3 days, 15 hours, 36 minutes
Three nines	99.9%	8 hours, 45 minutes, 36 seconds
Four nines	99.99%	52 minutes, 34 seconds
Five nines	99.999%	5 minutes, 15 seconds
Six nines	99.9999%	31.5 seconds
Seven nines	99.99999%	3.2 seconds
Eight nines	99.999999%	0.32 second
Nine nines	99.9999999%	0.03 second

**Gambar 3** Skala *The-Nines* dan Pengertiannya [16]

**3. Perancangan dan Implementasi**

Dalam proses perancangan sebuah sistem, diperlukan sebuah skenario yang terstruktur dengan baik. Untuk memudahkan proses perancangan implementasi diperlukan topologi yang membantu dalam memahami proses perancangan yang akan dibuat.



Gambar 4 Topologi Jaringan

Implementasi pembangunan dan perancangan sistem redundansi server, menggunakan *dedicated* server sebagai *master* server dan *cloud* server sebagai *backup* serta menggunakan Elastix sebagai server layanan VoIP untuk mendapatkan *high availability* server. *Dedicated* server dibangun berada dalam satu jaringan dengan cloud server yang akan diinstall pada Proxmox VE sebagai platform *cloud computing*. Selanjutnya client 1 dan client 2 akan berkomunikasi berupa layanan *voice call* dari *master* server kemudian akan mengalami kegagalan server atau *down*, sehingga selanjutnya kinerja akan diambil alih oleh *backup* server pada *cloud*. Selanjutnya digunakan pula *backup* server pada *dedicated* sebagai pembanding.

4. Pengujian dan Analisis

Pada bab ini akan dibahas mengenai pengujian dan analisis hasil implementasi yang telah dilakukan. analisis yang dilakukan bertujuan untuk mengetahui *availability* dari sebuah server yang dapat diketahui dari *downtime* serta ada *throughput* untuk mengetahui kualitas layanan. Pengujian dilakukan masing-masing pada *backup cloud* server dan *backup dedicated* server, dengan skenario sebagai berikut :

4.1 Pengukuran Downtime

Pengukuran *downtime* dilakukan untuk mengetahui saat server tidak dapat melayani client dikarenakan gagal server. Percobaan diukur pada saat master server mengalami *down* sehingga tidak dapat memberikan layanan, sampai *backup* server mengambil alih tugas tersebut (*failover*)Pengukuran *downtime* dilakukan dengan 2 skenario yakni saat *backup* pada *dedicated* server dan pada *cloud* server.

a. Sistematika Pengukuran

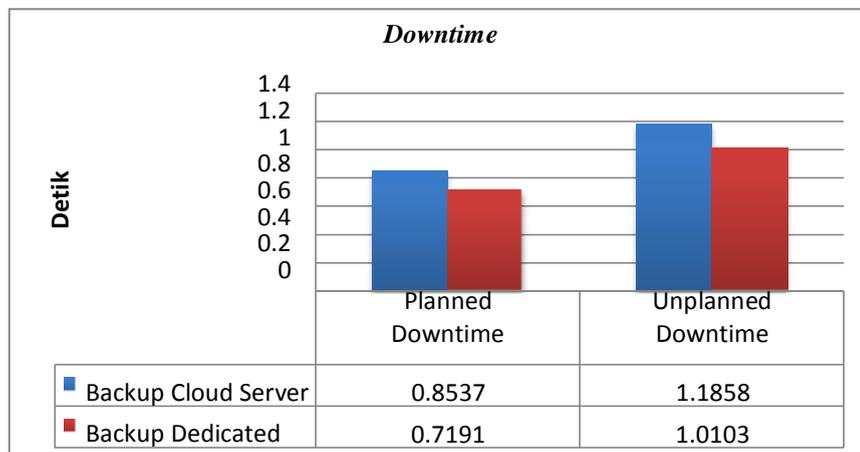
1. Downtime Planned

*Downtime planned* terjadi jika server memerlukan *maintenance* sehingga mengharuskan server tersebut dimatikan dengan sengaja. Pada percobaan ini dilakukan skenario dengan menggunakan perintah *init 0* ataupun *poweroff* yang menyebabkan server mati.

2. Downtime Unplanned

Pada *downtime unplanned* terjadi jika server mengalami kegagalan yang tidak terencana, seperti bencana alam. Pada skenario ini percobaan *downtime unplanned* dilakukan dengan cara memutuskan *link master* server sehingga server mengalami *down*.

b. Hasil Pengukuran



Gambar 5 Grafik Pengukuran Downtime

### c. Analisis Hasil Pengukuran *Downtime*

Hasil diatas menunjukkan bahwa nilai nilai *downtime* berkisar antara 0.7191 detik sampai 1.1858 detik, pada *unplanned downtime* lebih besar dikarenakan kegagalan yang terjadi bukan pada bagian server, melainkan kegagalan *link*, dimana paket ICMP yang membawa pesan dari *master server* ke *backup server* jika terjadi kegagalan tidak dapat diterima. Sehingga membutuhkan proses beberapa saat bagi *backup server* untuk mendeteksi bahwa *master server* sudah tidak dapat menerima layanan dari *client*. Sedangkan *planned downtime* terjadi kegagalan pada sisi server paket ICMP dapat sampai ke *backup server* dan memberitahu kepada *backup server* bahwa *master server* tidak dapat melayani *client*. Selanjutnya *backup server* siap untuk mengambil alih kinerja *master server*.

Pada perusahaan, kondisi *planned downtime* dapat terjadi karena masalah *maintenance* untuk *hardware*, *firmware*, maupun *software* [3] yang dapat terjadi 0-13 kali dalam setahun. Jika dilakukan perhitungan dengan menggunakan *backup server* dengan melihat rata-rata *downtime planned* dalam setahun adalah :

**Tabel 1** Tabel Pengukuran *Planned Downtime*

<i>Planned Downtime</i>		<i>Planned Downtime</i>	
<i>Backup Cloud Server</i>		<i>Backup Dedicated Server</i>	
1	x 0.8537 detik = 0.853667 detik/tahun	1	x 0.7191 detik = 0,719 detik/tahun
2	x 0.8537 detik = 1.707333 detik/tahun	2	x 0.7191 detik = 1.438187 detik/tahun
3	x 0.8537 detik = 2.561 detik/tahun	3	x 0.7191 detik = 2.15728 detik/tahun
4	x 0.8537 detik = 3.414667 detik/tahun	4	x 0.7191 detik = 2.876373 detik/tahun
5	x 0.8537 detik = 4.268333 detik/tahun	5	x 0.7191 detik = 3.595467 detik/tahun
6	x 0.8537 detik = 5.122 detik/tahun	6	x 0.7191 detik = 4.31456 detik/tahun
7	x 0.8537 detik = 5.975667 detik/tahun	7	x 0.7191 detik = 5.033653 detik/tahun
8	x 0.8537 detik = 6.829333 detik/tahun	8	x 0.7191 detik = 5.752747 detik/tahun
9	x 0.8537 detik = 7.683 detik/tahun	9	x 0.7191 detik = 6.47184 detik/tahun
10	x 0.8537 detik = 8.536667 detik/tahun	10	x 0.7191 detik = 7.190933 detik/tahun
11	x 0.8537 detik = 9.390333 detik/tahun	11	x 0.7191 detik = 7.910027 detik/tahun
12	x 0.8537 detik = 10.244 detik/tahun	12	x 0.7191 detik = 8.62912 detik/tahun
13	x 0.8537 detik = 11.09767 detik/tahun	13	x 0.7191 detik = 9.348213 detik/tahun
<b>Rata-rata 5.975667 detik/tahun</b>		<b>Rata-rata 5.033653 detik/tahun</b>	

Sedangkan untuk *unplanned downtime* dalam selang waktu rata-rata setahun adalah :

**Tabel 2** Tabel Pengukuran *Unplanned Downtime*

<i>Unplanned Downtime</i>		<i>Unplanned Downtime</i>	
<i>Backup Cloud Server</i>		<i>Backup Dedicated Server</i>	
1	x 1.1858 detik = 1.1858 detik/tahun	1	x 1.0103 detik = 1.0103 detik/tahun
2	x 1.1858 detik = 2.3716 detik/tahun	2	x 1.0103 detik = 2.020667 detik/tahun
3	x 1.1858 detik = 3.5574 detik/tahun	3	x 1.0103 detik = 3.031 detik/tahun
4	x 1.1858 detik = 4.7432 detik/tahun	4	x 1.0103 detik = 4.041333 detik/tahun
5	x 1.1858 detik = 5.929 detik/tahun	5	x 1.0103 detik = 5.051667 detik/tahun
6	x 1.1858 detik = 7.1148 detik/tahun	6	x 1.0103 detik = 6.062 detik/tahun
7	x 1.1858 detik = 8.3006 detik/tahun	7	x 1.0103 detik = 7.072333 detik/tahun
8	x 1.1858 detik = 9.4864 detik/tahun	8	x 1.0103 detik = 8.082667 detik/tahun
9	x 1.1858 detik = 10.6722 detik/tahun	9	x 1.0103 detik = 9.093 detik/tahun
10	x 1.1858 detik = 11.858 detik/tahun	10	x 1.0103 detik = 10.10333 detik/tahun
11	x 1.1858 detik = 13.0438 detik/tahun	11	x 1.0103 detik = 11.11367 detik/tahun
12	x 1.1858 detik = 14.2296 detik/tahun	12	x 1.0103 detik = 12.124 detik/tahun
13	x 1.1858 detik = 15.4154 detik/tahun	13	x 1.0103 detik = 13.13433 detik/tahun
<b>Rata-rata 8.3006 detik/tahun</b>		<b>Rata-rata 7.072333 detik/tahun</b>	

Jika dibandingkan dengan nilai *downtime* tanpa menggunakan *backup server* yakni 172.6 menit/tahun.[3] Nilai yang dilakukan menggunakan *backup server* terlihat perbedaan yang sangat signifikan. Hal ini membuktikan bahwa peranan menggunakan *backup server* memberikan pengaruh yang cukup besar untuk memperkecil *downtime* yang terjadi apabila terjadi kegagalan server.

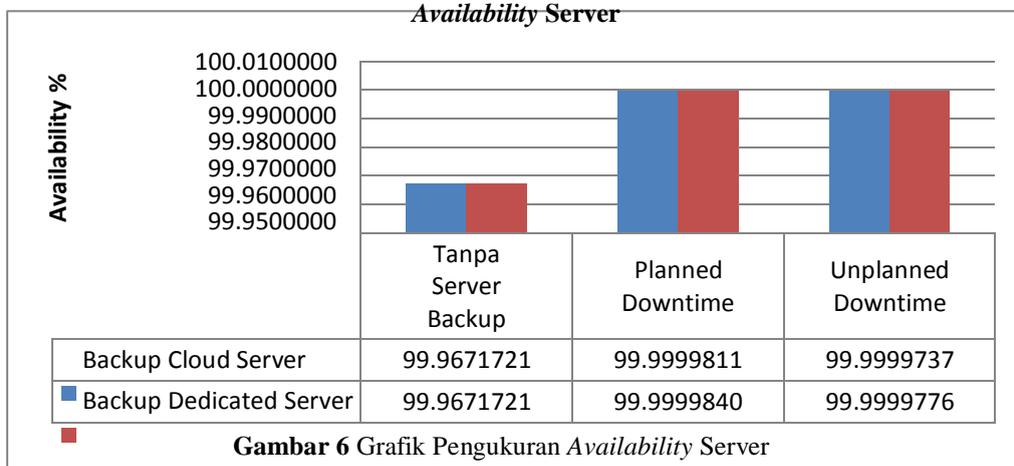
## 4.2 Perhitungan *Availability Server*

*Availability server* dilakukan untuk mengetahui karakteristik sistem dalam hal ketersediaan saat melayani *client*. Pada skenario ini akan dilakukan pada selang waktu pengamatan selama satu tahun menggunakan data survey [3].

**a. Sistematika Perhitungan**

Perhitungan dilakukan dengan mengambil data sebelumnya yakni nilai rata-rata *downtime* dalam selang waktu satu tahun menurut survey [3]. *Downtime* tersebut akan dibandingkan dengan jumlah *uptime* yang merupakan waktu yang dilakukan server untuk beroperasi dalam setahun. Pada percobaan kali ini akan dilakukan pengamatan *uptime* dalam selang satu tahun. Adapun perhitungan *availability* [14] telah dijelaskan pada bagian teori.

**b. Hasil Pengukuran**



**Gambar 6** Grafik Pengukuran *Availability* Server

**c. Hasil Analisis**

Dengan menggunakan persamaan seperti pada bagian teori [2] Pada perhitungan *availability* pada *planned downtime* di *backup server* baik pada *dedicated* dan *cloud* didapatkan hasil 99.9999% nilai tersebut masuk kedalam rekomendasi *the-six nines* [16] artinya bahwa setiap tahunnya sistem tersebut akan mengalami *down* atau tidak dapat melayani permintaan *client* dengan waktu tidak lebih dari 31,5 detik. Bila dibandingkan dengan tanpa menggunakan *backup server* pada skenario *planned* yang memiliki *availability* sebesar tidak lebih dari 8 jam, 45 menit, dan 36 detik adalah jauh lebih baik. Hal ini membuktikan bahwa peran menggunakan *backup server* untuk *handle* kondisi jika terjadi *down server* adalah sangat efektif.

Selain itu perhitungan diatas *downtime* dan *availability* dapat terlihat nilai yang dihasilkan saat *backup server* pada *cloud* dibandingkan pada *backup dedicated* lebih tinggi dikarenakan pada saat proses perpindahan (*failover*) ke *cloud server* paket harus melalui *hypervisor* yang menyebabkan paket akan tiba sedikit lebih lama. Sehingga *availability* yang dihasilkan pun berbeda namun masih masuk kedalam rekomendasi *the-six nines* sama seperti saat *backup* berada di *dedicated*.

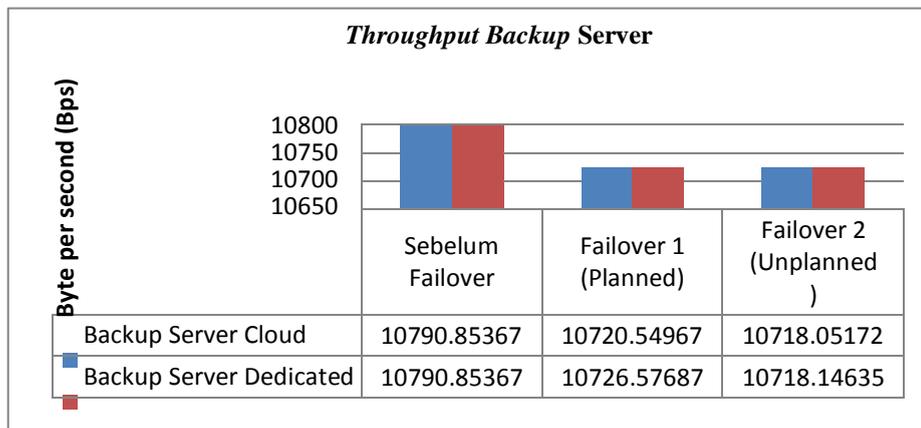
**4.3 Pengukuran Throughput**

Pengukuran *throughput* dilakukan untuk mengetahui kualitas layanan *voice call* saat sebelum dan setelah *failover* terjadi.

**a. Sistematika Pengukuran**

Skenario yang akan dilakukan adalah dengan cara mengukur *throughput* pada *failover* baik secara *planned* maupun *unplanned* yang selanjutnya akan dibandingkan dengan sebelum *failover*. Pengukuran dilakukan selama 1 menit dengan 30 kali percobaan menggunakan Wireshark.

**b. Hasil Pengukuran**



**Gambar 7** Grafik Pengukuran *Throughput* Server

### c. Hasil Analisis

Hasil grafik diatas menunjukkan *throughput* saat sebelum *failover* memiliki nilai yang paling tinggi, sedangkan untuk skenario *failover 2* atau *unplanned* mempunyai *throughput* yang paling rendah. Nilai *throughput* dipengaruhi oleh codec dan *bitrate* yang digunakan saat proses komunikasi. Pada percobaan kali ini dilakukan dengan menggunakan codec G.711 dengan *payload* 160 bytes dan *bitrate* 64kbps.

- Bitrate = 64 kbps
- Bitrate = payload x 8 x pps (G.711 memiliki 160 bytes)
- Pps =  $64k / (160 \times 8)$  pps = 50 pps
- IP Layer packet size = payload codec + RTP header + UDP header + IP header (IPv4)
- IP Layer Packet size =  $160 + 12 + 8 + 20 = 200$
- Frame Size (Ethernet Link) =  $200 + 14 = 214$  bytes (disebut juga total *packet size*)
- *Throughput* =  $214 \text{ bytes} \times 50 = 10700 \text{ bytes/sec}$  (*throughput* minimal yang harus sampai di penerima atau terbaca di wireshark)

Hasil analisis diatas menunjukkan nilai *throughput* keseluruhannya memiliki nilai diatas 10700 bytes/sec. Namun, perbedaan nilai terlihat ketika sebelum *failover* dan saat melakukan skenario *failover 1* dan *failover 2*. Penurunan *throughput* terjadi saat dilakukannya *failover*, hal ini dikarenakan saat terjadinya *failover* komunikasi tersebut akan terputus sehingga memerlukan proses *redial* untuk dapat berkomunikasi kembali, hal ini mengakibatkan paket yang sampai ditujukan dengan selang 1 menit akan lebih sedikit dibandingkan dengan sebelum *failover* yang tidak mengalami *drop call*.

Selanjutnya nilai *throughput* pada skenario *failover 1* memiliki nilai yang lebih besar karena memiliki waktu *down* yang lebih sedikit dibandingkan dengan *failover 2*, karena paket yang sampai ditujukan akan lebih banyak meskipun tidak jauh berbeda dengan *throughput* pada skenario *failover 2*.

## 5. Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan analisis dapat ditarik kesimpulan bahwa :

1. Implementasi server Elastix pada *dedicated* server dan *cloud* server untuk layanan VoIP berhasil dilakukan.
2. Pada saat terjadi *failover* proses komunikasi akan terputus, hal ini dikarenakan Elastix berbasis Asterisk yang bersifat B2BUA (*Back to Back User Agent*) jika terjadi gagal server akan melakukan terminasi, dan melakukan *signalling* ulang ketika server sudah siap seperti kondisi semula.
3. Hasil pengukuran *planned downtime* saat *backup* server pada *cloud* adalah 5.975667 detik/tahun, sedangkan tanpa menggunakan *backup* server dihasilkan 172.6 menit/tahun atau 10356 detik/tahun. Dengan begini implementasi menggunakan *backup* server pada *cloud* terbukti dapat mengurangi nilai *downtime*. Selanjutnya pada pengukuran *unplanned downtime* adalah 8.3006 detik/tahun
4. Hasil pengukuran *availability* dibandingkan dengan referensi survey untuk *planning downtime* yang meliputi segi *hardware*, *firmware* dan *software* yang memiliki *availability* server 99.9%, sedangkan pada implementasi ini memiliki nilai *availability* server sebesar 99.9999% baik untuk *planned downtime* maupun *unplanned downtime* yang memenuhi rekomendasi *the-six nines*.
5. Hasil pengukuran *throughput* saat sebelum *failover*, dan setelah *failover* bernilai stabil. Hal ini membuktikan bahwa meskipun terjadi *failover*, *throughput* yang didapat tidak mengalami perubahan yang signifikan sehingga kualitas layanan VoIP yang dirasakan tetap sama.

## DAFTAR PUSTAKA

- [1] A. Fathia, "Implementasi Dan Analisis Clustering Server Pada Integrasi Elastix Dan Openfire Untuk High Availability Server," Fakultas Elektro dan Komunikasi IT Telkom, Bandung, 2013.
- [2] C. D. Graziano, "A Performance Analysis of Xen and KVM Hypervisors for Hosting the Xen Worlds Project," Graduate Theses and Dissertations, Paper, 2011.
- [3] D. A. Kimber, X. Zhang, P. H. Franklin and E. J. Bauer, "Modeling Planned Downtime," Bell Labs Technical Journal, 2006.
- [4] E. V. Jain, "Failover Extension for Layer 2 Tunneling Protocol (L2TP) "failover"," RFC 4951, 2007.
- [5] F. Abidi and V. Singh, "Cloud Servers vs. Dedicated Server - A Survey," 2013.
- [6] I. Red Hat, "Delivering High Availability Solutions with Red Hat Cluster Suite," Revision 3c, 2003.
- [7] J. Dantas, R. Matos, J. Araujo and P. Maciel, "An Availability Model for Eucalyptus Platform: An Analysis of Warm-Standy," IEEE International Conference on System, Man, and Cybernetics, 2012
- [8] J. Gifari, "Analisis dan Implementasi Keamanan Layanan VoIP Based on Cloud Menggunakan Secure Realtime Transport Protocol dan Transport Layer Security," Fakultas Teknik Elektro, Universitas Telkom, Bandung, 2014.
- [9] J. Lee and K. Lee, "SYNCEYE: An Availability Measurement Tool for Embedded System," Asia-Pacific Software Engineering Conference, 2014.
- [10] Linux-KVM, "Kernel Virtual Machine," 2014. [Online]. Available: [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page). [Accessed 29 Mei 2015].
- [11] Microsoft, "Failover Cluster," 2015. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff650328.aspx>. [Accessed 25 April 2015].
- [12] P. Mell and T. Grance, "The NIST Definiton of Cloud Computing," NIST Special Publication, p. 7, 2011.
- [13] "Quality of Service (QoS)," Cisco, [Online]. Available: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html>. [Accessed 13 April 2015].
- [14] R. Munadi, Teknik Switching, Bandung: Informatika, 2011.
- [15] S. Corporation, "Veritas Cluster Server User's Guide," 2006.
- [16] S. Snedaker, Best Damn Windows Server 2003, Canada: Syngress, 2003.
- [17] VMWare, "Understanding Full Virtualization, Paravirtualization, and Hardware Assisted," White Paper, p.14, 2007.
- [18] W. Lau, "A Comprehensive Intoduction to Cloud Computing," 16 December 2011. [Online]. Available: <https://www.simple-talk.com/cloud/development/a-comprehensive-introduction-to-cloud-computing/>. [Accessed 12 April 2015]
- [19] I. Cloudhost, "Dedicated Server," [Online]. Available: <https://idcloudhost.com/wp-content/uploads/2015/01/Server-982x10242-282x300.png>. [Accessed 3 Juni 2015].
- [20] P.B.-T.I. Company. [Online]. Available: [https://www.profitbricks.com/sites/default/files/slide\\_03\\_02\\_transparent\\_bg.png](https://www.profitbricks.com/sites/default/files/slide_03_02_transparent_bg.png). [Accessed 3 Juni 2015].
- [21] M. S. Intacore. [Online]. Available: [http://www.intacore.com/workspace/uploads/page\\_images/voip\\_diagram-4f13441b9760b.jpg](http://www.intacore.com/workspace/uploads/page_images/voip_diagram-4f13441b9760b.jpg). [Accessed 3 Juni 2015].