

IMPLEMENTASI DAN ANALISIS CLUSTERING SERVER PADA INTEGRASI ELASTIX DAN OPENFIRE UNTUK HIGH AVAILABILITY SERVER

Fathia Amany¹, Rendy Munadi², Iman Hedi Santoso³

¹Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

Seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi, kebutuhan manusia akan komunikasi realtime tak dapat terelakkan lagi. Dalam konteks sistem Informasi/Komunikasi manusia sebagai client membutuhkan server sebagai penyedia layanan yang handal (highly available). Ketersediaan server untuk melayani kebutuhan client selama 24 jam menjadi sangat krusial. Pada kenyataannya terdapat banyak kendala dalam mewujudkan kondisi high availability tersebut. Maka diperlukan solusi tepat untuk mengatasi kemungkinan sistem akan mengalami gangguan baik dari luar maupun dari dalam yang dapat menyebabkan gagal server. Teknik Server Cluster dapat menjadi solusi untuk mewujudkan kondisi high availability tersebut. Teknik Server Cluster ini akan mengelompokkan beberapa server menjadi satu cluster yang terdiri dari server primer dan server sekunder serta menerapkan sistem redundant / backup dengan metode failover pada sistem tersebut. Server backup ini dapat menggantikan kinerja server yang mengalami kegagalan dalam melayani kebutuhan client.

Dalam Tugas Akhir ini akan diimplementasikan teknik Server Cluster pada Server Elastix dan Openfire sebagai server primer. Sehingga ketika salah satu server primer yang terdapat pada cluster tersebut mengalami gagal server, server lain akan bertindak sebagai backup dan menggantikan kinerja server primer untuk mewujudkan kondisi high availability.

Dari hasil pengukuran didapat bahwa nilai downtime 0.692 detik untuk perpindahan dari server alpha ke server beta, 1.622 detik untuk perpindahan dari server beta ke server charlie, serta 1.701 untuk perpindahan dari server alpha ke server charlie. Implementasi server cluster ini sangat efektif dalam mengurangi nilai downtime apabila tanpa menggunakan server cluster yang harus menunggu waktu reboot selama 2.4 menit. Nilai availability yang didapatkan berdasarkan hasil perhitungan adalah 99.999% , telah memenuhi rekomendasi "The Five-nines". Hasil pengukuran QoS yang dilakukan masih memenuhi standar "baik", yaitu delay = < 150ms, jitter = < 50ms (ITU), standar "sedang" untuk packet loss = < 15% (Tiphon). Hasil pengukuran MOS untuk layanan instant messaging masih memenuhi standar "baik" 4,1 - 4,9 (ITU-T P.800) dan standar "cukup baik" 3,1-3,9 untuk layanan voice dan video call. Sistem ini memenuhi kualitas layanan VoIP sehingga layak untuk diimplementasikan.

Kata Kunci : realtime, high availability , server cluster, server primer, server backup, failover

Telkom
University

Abstract

Due to the rapid growth of information and communication technology, the need of real-time communication is also keep increasing. In the context of communication or information system, human as a client needs a high-availability server for service provider. The existence of server to serve clients for 24 hours is very crucial. In fact, there are plenty of constraint to keep that available. Hence, we need a solution to overcome the possibility of server failure, by both of inside or outside cause. Server cluster technique is the way out for that highly available server. Server cluster technique classifies several servers into one cluster, consists of primary and secondary servers and apply redundant system. These backup servers cover the server's failure in providing the service for clients.

This final project implements the server cluster technique on Elastix and Openfire as primary server. So when one of the servers within the cluster fails, another server will act as a backup to keep it highly available.

The result of downtime measurement is taking 0.692 second to move from alpha server to beta server, and taking 1.622 second to move from beta server to charlie server, and taking 1.701 second to move from alpha server to charlie server. This implementation is highly effective for decreasing the downtime (without server cluster it takes 2.4 minutes). Availability score is obtained based on the result of the calculation is 99.999%, have met "The Five-nines" recommendation. QoS score for jitter are standardized as "Good" (based on ITU-T Standard) and "Medium" for packet loss scores (based on Tiphon Standard). MOS measurement result for instant messaging service have met the "good" standard which is range in 4.1-4.9 (based on ITU-T P.800) and for voice and video calls have met the "good enough" standard which is range in 3.1-3.9 (based on ITU-T P.800). This overall system is considered good VoIP quality, so it is feasible to be implemented.

Keywords : high availability , server cluster, primary server, backup server, failover.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan meningkatnya kebutuhan manusia akan komunikasi yang bersifat *realtime*, perkembangan teknologi yang ada saat ini dituntut untuk dapat memberi solusi untuk mengatasi masalah-masalah yang timbul, salah satunya adalah ketersediaan *server* untuk dapat melayani kebutuhan dari pengguna layanan atau *client* atau yang biasa disebut *high availability server*. Manusia sebagai *client* dalam hal ini membutuhkan teknologi yang sangat praktis, efektif dan efisien. Informasi tidak hanya dapat diakses kapanpun dan dimanapun melainkan dalam berbagai kondisi. *Server* yang melayaninyapun harus memiliki kemampuan untuk melayani pengguna dalam berbagai kondisi dengan kualitas layanan yang baik sepanjang waktu. Pada kenyataannya terdapat banyak gangguan bagi *server* dalam melayani berbagai kebutuhan *client*. Gangguan tersebut dapat bermacam-macam seperti rusaknya kondisi fisik perangkat dan banyaknya trafik yang melebihi kemampuan *server*. Hal ini dapat menyebabkan terjadinya *server down* atau gagal *server* sehingga tidak dapat melayani *client* sepanjang waktu.

Teknik *Server Cluster* memiliki kemampuan untuk meningkatkan *high availability server* atau ketersediaan *server* untuk dapat melayani kebutuhan *client* sepanjang waktu. Apabila terdapat gangguan yang menyebabkan salah satu *server down* maka dilakukan proses *failover* yaitu pengambilalihan kerja oleh *server* lain yang berada di dalam *cluster* tersebut agar *client* dapat terus dilayani tanpa adanya gangguan.

Oleh karena itu, dengan diimplementasikannya teknik *Server Cluster* pada Tugas Akhir ini diharapkan mampu mewujudkan kondisi *high availability server* dan memberikan jaminan QoS pada layanan yang diberikan.

1.2 Tujuan

Tujuan penelitian pada tugas akhir ini adalah :

1. Memahami konsep dasar *Server Cluster*.
2. Mengimplementasikan konsep *failover* pada sebuah *server cluster*.
3. Memperkecil nilai *downtime*.
4. Mewujudkan kondisi *high availability server*.

1.3 Rumusan Masalah

Permasalahan yang dijadikan objek penelitian dan pengembangan tugas akhir ini adalah :

1. Bagaimana mengintegrasikan sever Elastix dan Openfire untuk membuat layanan *triple play*.
2. Membangun sebuah *server cluster* menggunakan *software Heartbeat* dan DRBD.
3. Mengimplementasikan *server cluster* terhadap *server* Elastix dan Openfire untuk mewujudkan kondisi *high availability server*.
4. Menganalisis hasil implementasi yang telah dilakukan dengan mengukur parameter-parameter *availability* dan *downtime* serta nilai QoS dari layanan yang diberikan.

1.4 Batasan Masalah

Batasan masalah dalam penelitian dan pengembangan tugas akhir ini adalah:

1. Layanan yang diimplementasikan adalah layanan *voice call*, *video call* dan *instant messaging*.
2. *Server VoIP* yang digunakan adalah *server* Elastix.
3. *Server IM* yang digunakan adalah *server* Openfire.
4. Jaringan LAN yang akan dibangun adalah jaringan VoIP dengan tiga buah *server* dalam satu *cluster* yaitu *server alpha*, *beta* dan *charlie*, dua buah *client* dan dua buah *switch* sebagai penghubung.
5. Pengamatan QoS akan dilakukan selama 5 menit.
6. Skenario yang akan dilakukan terdiri dari tiga kali *failover* yaitu *failover 1* (perpindahan dari *server alpha* ke *beta*), *failover 2* (perpindahan dari *server beta* ke *charlie*), dan *failover 3* (perpindahan dari *server alpha* ke *charlie*).
7. Percobaan dan pengambilan data dilakukan sebelum, sesudah *failover 1* , sesudah *failover 2* dan sesudah *failover 3*.
8. Percobaan dan pengambilan data dilakukan di gedung E Institut Teknologi Telkom
9. Parameter *availability server* yang akan dianalisis adalah *availability* dan *downtime server*.
10. Parameter QoS yang dianalisis meliputi *delay*, *jitter*, *troughput* dan *packet loss* untuk layanan *voice* dan *video call*, lalu parameter *round trip time* untuk layanan *instant messaging*.

1.5 Metode Penelitian

Metode yang akan digunakan untuk menyelesaikan Tugas Akhir ini adalah :

a. Studi literatur

Studi literatur ini dimaksudkan untuk mempelajari konsep dan teori- teori yang dapat mendukung proses perancangan sistem.

b. Perancangan dan realisasi

Meliputi aplikasi dari konsep dan teori yang telah diperoleh. Melakukan pengujian terhadap hasil perancangan yang telah dikerjakan.

c. Pengujian dan analisis implementasi

Menganalisis nilai *downtime*, *availability* dan QoS hasil implementasi *Server Cluster* pada integrasi Elastix dan Openfire

1.6 Sistematika Penulisan

Secara umum sistematika penulisan Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi uraian singkat mengenai latar belakang permasalahan, tujuan penelitian, batasan masalah, metode penelitian serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi penjelasan tentang dasar – dasar teori yang berkaitan dengan arsitektur jaringan yang akan dirancang sehingga akan membantu pengerjaan Tugas Akhir ini.

BAB III PERANCANGAN SISTEM DAN IMPLEMENTASI

Pada bab ini akan dijelaskan diagram alir (*flowchart*) pengerjaan serta perancangan dimulai dari deskripsi masalah dan skenario pengumpulan data.

BAB IV PENGUJIAN DAN ANALISIS HASIL IMPLEMENTASI

Evaluasi dan analisis hasil dari penelitian dibahas disini, Beserta analisis beberapa parameter yang berhasil dicapai.

BAB V PENUTUP

Berisikan kesimpulan dari hasil pengukuran yang telah dilakukan, serta rekomendasi atau saran untuk perbaikan dan pengembangan lebih lanjut.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

1. Implementasi teknik *server cluster* untuk *server alpha*, *beta* dan *charlie* berhasil dilakukan.
2. Integrasi antara *server Elastix* untuk layanan VoIP dengan *server Openfire* untuk layanan *instant messaging* berhasil diimplementasikan.
3. Pada saat terjadi *failover* proses komunikasi akan terputus, hal ini dikarenakan *server Elastix* berbasis Asterisk sebagai IPPBX-nya yang memiliki sifat *Back to Back User Agent*. Sehingga apabila terjadi gagal *server* akan melakukan terminasi kemudian melakukan proses *signaling* ulang ketika *server* telah kembali dalam kondisi semula.
4. Hasil pengukuran *downtime* yang dilakukan memiliki nilai 0.692 detik untuk perpindahan dari *server alpha* ke *server beta*, 1.622 detik untuk perpindahan dari *server beta* ke *server charlie*, serta 1.701 untuk perpindahan dari *server alpha* ke *server charlie*. Nilai *downtime* yang didapatkan telah memenuhi rekomendasi *downtime* pertahun yaitu 0.88 jam atau 3168 detik setiap tahunnya. ^[13] Implementasi *server cluster* ini sangat efektif dalam mengurangi nilai *downtime* apabila tanpa menggunakan *server cluster* yang harus menunggu waktu *reboot* selama 2.4 menit.
5. Hasil pengukuran *availability* yang dilakukan berdasarkan survey *infonetics.com* yang dibandingkan dengan nilai *downtime* yang didapat pada pengukuran memiliki nilai 99.999%.
6. Hasil pengukuran QoS yang dilakukan masih memenuhi standar "baik", yaitu *jitter* = < 50ms (ITU), standar "sedang" untuk *packet loss* = < 15% (Tiphon).
7. Hasil pengukuran MOS yang dilakukan pada 30 responden untuk layanan *instant messaging* masih memenuhi standar "baik" 4,1 - 4,9 (ITU-T P.800) dan standar "cukup baik" 3,1-3,9 untuk layanan *voice* dan *video call*.

5.2 Saran

Saran yang dapat diajukan untuk penelitian lebih lanjut mengenai topik ini adalah :

1. Perlu dilakukan penelitian lebih lanjut dalam menanggulangi komunikasi yang terputus pada saat proses *failover* terjadi.
2. Perlu dilakukan penelitian lebih lanjut untuk melakukan *automatic redial* di sisi pengguna saat terjadi *drop call*.
3. Perlu dilakukan skenario *failover* lain seperti *symmetric failover* untuk dapat melayani berbagai macam layanan dalam waktu yang bersamaan.
4. Perlu dilakukan penelitian lebih lanjut mengenai *Load Balancing Cluster* untuk melakukan distribusi pekerjaan ketika terjadi *overload* pada salah satu *server*.
5. Perlu dilakukan penelitian lebih lanjut mengenai pengukuran *end to end delay*.
6. Perlu dilakukan penelitian lebih lanjut pada *cluster manager* selain Heartbeat yaitu OpenAIS dan Pacemaker.



Telkom
University