IMPLEMENTASI FAILOVER CLUSTERING PADA DUA PLATFORM YANG BERBEDA UNTUK MENGATASI KEGAGALAN FUNGSI SERVER

Akhyar Muchtar
Lecturer of Bosowa Polytechnic

Abstrak

Penelitian ini bertujuan untuk (1) membuat sebuah rancangan failover clustering sebagai salah satu solusi dalam menangani masalah kegagalan fungsi server. (2) membandingkan performance sistem failover pada sistem operasi Linux dan Windows. Metode penelitian yang digunakan adalah metode penelitian experimental, yaitu dengan melakukan pengujian langsung pada masing-masing teknologi berdasarkan parameter tertentu dan kondisi yang terkendali. Hasil penelitian menunjukkan bahwa dalam perancangan penerapan failover clustering pada Proxmox sebagai salah satu Distro Linux dengan menggunakan UCARP dan DNS failover sama baiknya dari segi kinerja dalam mencapai nilai standar availability, namun DNS failover sedikit lebih baik karena dapat digunakan pada platform Linux ataupun Windows. Dalam penelitian ini juga diketahui failover cluster berbasis Linux dengan menggunakan UCARP ataupun DNS failover dan teknologi failover yang terdapat di Windows memproleh tingkat availability yang sama dan memenuhi standar minimum fault tolerance yang telah ditetapkan yakni sebesar 99,99%.

Kata kunci :*Kegagalan server, Failover, Cluster, Fault Tolerance, UCARP, DNS Failover.*

Abstract

This research aimstoprovideadraftfailoverclusteringas onesolutionto addressing the problemservermalfunction. In addition,this research also compared the performance ofthe system fail over on Linux and Windows operating systems. The research usedis the experimental research method by conducting directtestingon each technology based on parameters and controlled conditions.The results showedthatin the design of application fail over clusteringon Proxmoxas one of the Linux distributions using DNS fail over UCARP and just as good in termsof performancein achievingthe standards of availability, but the DNS fail overs lightly better because itcan be used on Linux or Windows platform. In the present studyis also known Linux-based cluster fail over using DNS failoverand UCARPor fail over technologies containedin Windows farethe same level of availability and fault toleranceto meet minimum standards setwhich is equal to 99.99%.

Kata Kunci *: Failure ofserver, Failover, Cluster, Fault Tolerance, UCARP, DNS Fail over.*

A. PENDAHULUAN

Salah satu permasalahan penting yang dihadapi oleh sebuah perusahaan atau instansi yang berskala besar maupun menengah adalah tersedianya infrastruktur teknologi informasi yang kuat dan memadai dalam mengelola ribuan bahkan jutaan data penting setiap harinya.Salah satu infrastruktur yang digunakan dalam mengelola data-data tersebut adalah *server*(Garnieri,

2010).server merupakan sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer (Komaruddin, 2012). Mengingat fungsi yang dimiliki *server* adalah memberikan layanan kepada *client*, maka *server*dituntut untuk seminimal mungkin mengalami gangguan yang dapat menggangu layanan yang diberikan kepada *client.Server* sendiri merupakan sebuah perangkat yang adakalanya mengalami kerusakan sehingga diperlukan penanganan tertentu untuk mengatasi masalah tersebut.

Pelayanan *server* misalnya pada sebuah institusi pendidikan ataupun organisasi lain seperti perusahaan, tentunya telah memiliki infrastruktur yang memadai dalam memberikan informasi yang lengkap kepada *client*nya. Ketika masyarakat sebagai *client* ingin memperoleh informasi tentang institusi ataupun organisasi tersebut, mereka dapat langsung mengakses informasi yang telah disediakan pada sistem yang dimiliki oleh institusi ataupun organisasi tersebut, namun terkadang pada saat sistem informasi tersebut diakses, terjadi kegagalan.Hal ini disebabkan karena pada *server* sistem informasi, terjadi *failure* atau kegagalan.Kegagalan ini disebabkan karena *server* utama mati dan tidak ada server *backup* yang menggantikan fungsi *server* utama yang mati.

Salah satu solusi untuk mengatasi masalah di atas adalah dengan menggunakan teknologi *failover clustering*.Untuk menggunakan teknologi ini maka dibutuhkan minimal dua server yang digabungkan dalam satu cluster.Cluster (Shimonski, 2003)adalah sekelompok mesinyang bertindak sebagai sebuah entitas tunggal untuk menyediakan sumber daya dan layanan ke jaringan.Sedangkan*failoverclustering*(Hirt, 2009) bertujuan untuk membantu menjaga akses client ke aplikasi dan sumber daya server, bahkan ketika terjadi kegagalan software, ataupun kegagalan fungsi *server*yang mengakibatkan server berhenti bekerja.. Teknologi ini diharapkan dapat menjadi solusi dalam mengatasi kegagalan *server* ketika terjadi gangguan ataupun perawatan (*maintenance*).

Berdasarkan hal tersebut, maka pada penelitian ini, penulis berencana mengimplementasikan *failover clustering* dan membandingkan *failover clustering* pada dua *platform* yang berbeda (Windows dan Linux). Dalam Hal ini, diharapkan akan diketahui nilai *downtime*

minimum yang dihasilkan ketika terjadi proses *fileover* pada sistem.

B. METODE PENELITIAN

Pada penelitian ini, digunakan metode penelitian eksperimental, yakni metode penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dalam kondisi yang terkendali (Dharma. 2008). Pengaruh perlakuan yang dimaksud adalah pengaruh penggunaan masing-masing teknologi teknologi *failover clustering* terhadap availability server. Masing-masing teknologi akan diuji coba pada mesin yang sama, kemudian dilakukan pengukuran availability dengan memberi gangguan pada server. Hal ini dilakukan secara berulang-ulang, kemudian dilakukan perbandingan berdasarkan parameter yang diperoleh saat pengukuran.

Pengukuran Performance Server Stand Alone

Arsitektur Sistem

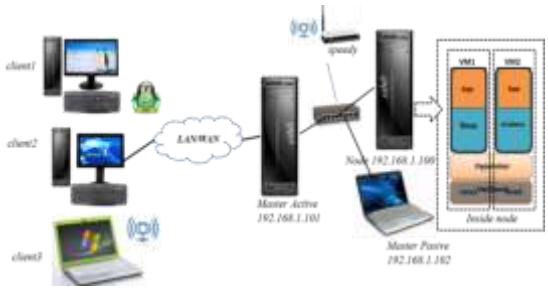
Untuk arsitektur pada system *stand alone*, hanya menggunakan satu buah server dan satu client yang mengakses server tersebut. Pada server dapat instalkan system operasi Ubuntu server 10.10 ataupun Windows Server 2008 R2,kemudian dikoneksikan ke jaringan.Pengujian yang dilakukan pada server *stand alone* mencakup pengujian kinerja server, yakni *availability* server dan maksimum *workload* server. Pengujian *availability* dilakukan dengan cara memberikan gangguan pada server berupa memutuskan sumber daya server ataupun jaringan pada server. Sedangkan pengujian *workload* server dilakukan dengan menggunakan *benchmarking tool*, yakni menggunakan httperf.Testing dilakukan dengan memasukkan perintah pada httperf untuk mengirimkan HTTP *request* sebanyak 500 *request* ke server. Pengujian lain adalah pengujian kinerja jaringan, yakni *latency* dan paket loss. Cara pengujiannya adalah dengan melakukan "*ping"* dari *client* ke server, sambil memberi gangguan pada server. Dari hasil ping dapat diketahui besar *latency (delay)* dan *packet loss*.

Performance Failover Windows 2008 R2

Arsitektur Sistem

Arsitektur system yang digunakan secara umum dapat dilihat pada**gambar 1,**yang digunakanadalah tiga buah server dan tiga buah client dimana posisi client diasumsikan dapat berada dimana saja.Pada masing-masing server dikonfigurasi Windows Server 2008 R2 dengan memilih *data centreinstallation*.

Pengujian yang dilakukan pada server *failover cluster* dengan *platform Windows Server 2008 R2*, sama dengan pengujian yang dilakukan pada server *stand alone,* yaknimencakup pengujian kinerja server, yakni *availability* server dan maksimum *workload* server. Pengujian *availability* dilakukan dengan cara memberikan gangguan pada server berupa memutuskan sumber daya server ataupun jaringan pada server. Sedangkan pengujian *workload* server dilakukan dengan menggunakan *benchmarking tool*, yakni menggunakan httperf.Testing dilakukan dengan memasukkan perintah pada httperf untuk mengirimkan HTTP *request* sebanyak 500 *request* ke server. Pengujian lain adalah pengujian kinerja jaringan, yakni *latency* dan paket loss. Cara pengujiannya adalah dengan melakukan "*ping"* dari *client* ke server, sambil memberi gangguan pada server. Dari hasil ping dapat diketahui besar *latency (delay)* dan *packet loss*.



Gambar 1. Rancangan *failover* secara umum

Pengukuran Performance Failover Pada Proxmox VE 2.1 Menggunakan UCARP

Arsitektur Sistem

Pada **gambar 1** digunakan tiga buah server.digunakan tiga buah serverdan tiga buah client dimana posisi client diasumsikan dapat berada dimana saja. Pada masing-masing server diinstalkan *Oprting system* Proxmox VE 2.1, kemudian pada server Master dan server slave diinstalkan *software* aplikasi UCARP. UCARP (*Using Common Address Redundancy Protocol*) atau CARP merupakan protocol yang memungkinkan beberapa host yang terdapat pada satu jaringan untuk berbagi satu alamat ip virtual yang sama (Laurent, 2012). Pengujian yang dilakukan pada server *failover cluster* dengan *platform Linux*, yakni *Proxmox VE,* sama dengan pengujian yang dilakukan pada server *stand alone,* yaknimencakup pengujian kinerja server, yakni *availability* server dan maksimum *workload* server. Pengujian *availability* dilakukan dengan cara memberikan gangguan pada server berupa memutuskan sumber daya server ataupun jaringan pada server. Sedangkan pengujian *workload* server dilakukan dengan menggunakan *benchmarking tool*,

yakni menggunakan httperf.Testing dilakukan dengan memasukkan perintah pada httperf untuk mengirimkan HTTP *request* sebanyak 500 *request* ke server. Pengujian lain adalah pengujian kinerja jaringan, yakni *latency* dan paket loss. Cara pengujiannya adalah dengan melakukan "*ping"* dari *client* ke server, sambil memberi gangguan pada server. Dari hasil ping dapat diketahui besar *latency (delay)* dan *packet loss*.

Pengukuran Performance Virtual Virtual Ubuntu Server 10.10 Maveric Merkat

Arsitektur Sistem

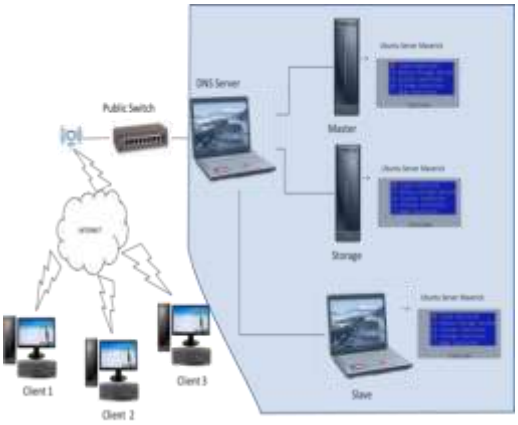
Arsitektur yang digunakan pada system ini kurang lebih sama dengan arsitektur jaringan (**gambar 1**)yangdigunakan pada pengukuran *performace* pada Proxmox VE 2.1.perbedaannyahanyalah konfigurasi UCARP dilakukan pada mesin server virtual yang terdapat pada server Mynode. Server vitual yang dinstalkan adalah dua buah server virtual Ubuntu Server 10.10 Maverick Merkat. Didalam server vitual tersebut kemudian dilakukan perlakuan yang sama pada mesin fisik di pengukuran performance pada Proxmox VE 2.1 dimana di instalkan UCARP dan dikonfigurasi memiliki IP virtual yang sama, namun salah satunya di set sebagai *master* dan yang satunya lagi di set ebagai *slave*. Pengujian yang dilakukan pada mesin virtual *Ubuntu* server 10.10, sama dengan pengujian yang dilakukan pada server *stand alone*, yaknimencakup pengujian kinerja server, yakni *availability* server dan maksimum *workload* server. Pengujian *availability* dilakukan dengan cara memberikan gangguan pada server berupa memutuskan sumber daya server ataupun jaringan pada server. Sedangkan pengujian *workload* server dilakukan dengan menggunakan *benchmarking tool*, yakni menggunakan httperf.Testing dilakukan dengan memasukkan perintah pada httperf untuk mengirimkan HTTP *request* sebanyak 500 *request* ke server. Pengujian lain adalah pengujian kinerja jaringan, yakni *latency* dan paket loss. Cara pengujiannya adalah dengan melakukan "*ping"* dari *client* ke server, sambil memberi gangguan pada server. Dari hasil ping dapat diketahui besar *latency (delay)* dan *packet loss*.

Pengukuran Performance DNS Failover Pada Ubuntu Server 10.10

Arsitektur sistem

Pada **gambar 2** digunakan empat buah server kemudian salah satu sever digunakan sebagai DNS server.

Implementasi DNS server dapat diujicobakan pada dua server yang berbeda sytem oprasinya namun untuk pengukuran performance pada penelitian ini pada masing-masing server diinstalkan Ubuntu Server 10.10 (yakni untuk cloud controller dan cluster controller) dan tiga buah client. Pada saat penginstalan Ubuntu Server 10.10 *Maveric Merkat*untuk mengaktifkan layanan cloud computing pilih *Ubuntu Enterprise Cloud installation*. Pengujian yang dilakukan *Linux Ubuntu* server 10.10, sama dengan pengujian yang dilakukan pada server *stand alone*, yaknimencakup pengujian kinerja server, yakni *availability* server dan maksimum *workload* server. Pengujian *availability* dilakukan dengan cara memberikan gangguan pada server berupa memutuskan sumber daya server ataupun jaringan pada server. Sedangkan pengujian *workload* server dilakukan dengan menggunakan *benchmarking tool*, yakni menggunakan httperf.Testing dilakukan dengan memasukkan perintah pada httperf untuk mengirimkan HTTP *request* sebanyak 500 *request* ke server. Pengujian lain adalah pengujian kinerja jaringan, yakni *latency* dan paket loss. Cara pengujiannya adalah dengan melakukan "*ping"* dari *client* ke server, sambil memberi gangguan pada server. Dari hasil ping dapat diketahui besar *latency (delay)* dan *packet loss*.



Gambar 2. Arsitektur Jaringan *Failover Cluster* menggunakan DNS *failover* pada Ubuntu Server 10.10 Maveric Merkat

Gambar 3. Konfigurasi UCARP

C. HASIL

Pengukuran Performance Server Stand Alone

Berdasarkan hasil pengukuran pada table 1, terlihat bahwa server *stand alone* mengalami gangguan selama 7 jam, 34 menit, dengan waktu pengukuran 24 jam, sehingga waktu *uptime*-nya adalah 16 jam 25 menit, dari hasil tersebut nilai *availability* yang dicapai sebesar 68,403%.Hal ini tentu jauh dari nilai standar availability yang diizinkanyaitu sebesar 99,99% (Anonim, 2010). Dari hasil yang diperoleh juga dapat diketahui, bahwa dengan menggunakan satu server yang terus menerus melayani permintaan dari client akan sangat tidak efektif ketika terjadi gangguan pada server tersebut. Hal ini dikarenakan saat terjadi gangguan pada server, layanan pada server tersebut akan berhenti dan hanya akan berfungsi kembali ketika gangguan pada server sudah ditangani atau di *maintenance*.Hasil pengukuran ini digunakan hanya untuk membandingkan kondisi server sebelum diterapkan *system failover* maupun setelah diterapkan *system failover*, sehingga dapat diketahui sejauh mana *sytem failover* dapat menjamin ketersedian layanan ketika terjadi gangguan pada server utama.

Pengukuran Performance Failover Windows 2008 R2

Berdasarkan hasil pengukuran pada table 1 di atas, *failover cluster*pada Windows Server 2008 R2 memenuhi standar minimum *fault tolerance* yang telah ditetapkan, yakni sebesar 99.99 % (Anonim, 2010).Failover *cluster* pada windows relative stabil karena pada Windows server 2008 R2 sendiri, *failover cluster* merupakan salah satu fasilitas dari windows. Jadi sudah include dalam fasilitas *future add ons*. *Failover Cluster* ini hanya mendukung untuk aplikasi SQL Server, aplikasi yang lain tidak di*cover*. Oleh karena itu, peran NAS sangat dominan. NAS juga hanya men*cover* masalah file, bukan aplikasi. Jadi NAS hanya menyediakan storage yang fungsinya untuk menampung file dari sistem operasi apapun. Saat terjadi kegagalan fungsi server, maka server yang satu langsung mengambil alih dengan data mengambil pada NAS.

Pengukuran Performance Failover Pada Proxmox VE 2.1 Menggunakan UCARP

Pada Proxmox versi 2.0 ke bawah, kegagalan fungsi server seperti *failover* maupun *loadbalancing* tidak ada, kecuali dikonfigurasi secara tersendiri. Sedangkan untuk versi 2.0 ke atas, sudah ada package untuk mengatasi kegagalan

fungsi server, yakni HA *cluster* (*High Availability Cluster)*, namun tetap harus dilakukan konfigurasi tambahan untuk mengaktifkan *Fencing* serta membutuhkan peralatan tambahan berupa *switched rack* PDU (*Power Distribution Unit*).

Pada prinsipnya HA pada proxmox menggunakan prinsip *live migration*, dimana terlebih dahulu ditentukan mesin virtual yang akan dibackup jika terjadi kegagalan pada sisi server. Jadi jika terdapat 9 virtual mesin, maka ke sembilan virtual mesin ini masing-masing dikonfigurasi untuk live migration dengan meng-*enablekan* HA.

Jika mesin virtual sudah dikonfigurasi untuk HA, maka diperlukan sinkronisasi antara mesin utama dengan node yang akan menjadi tempat migrasi dari mesin virtual tersebut. Sinkronisasi ini akan berjalan dengan baik jika *fencing* sudah dikonfigurasi. *Fencing* hanya akan berjalan jika ada switched rack PDU *(Power Distribution Unit)*, yakni perangkatdilengkapi denganbeberapa outputdirancanguntuk mendistribusikantenaga listrik,terutama untukrakkomputerdan peralatan jaringanyang terletak di dalamsebuah pusat data.Khusus untuk penerapan package yang sudah tersedia di proxmox pengujiannya ini tidak dilakukan karena keterbatasan peralatan sehingga untuk mengatasi hal tersebut, maka dirancang sistem jaringan seperti pada gambar 6.

Pada mesin server yang telah diinstalkan proxmox, diterapkan *simple failover* menggunakan UCARP. Dengan menggunakan UCARP, *availability* server bisa mencapai nilai *fault tolerance*, yakni 99,99%. Prinsip kerja dari UCARP yaitu memungkinkanbeberapa hostuntuk berbagialamatIPumumvirtual untukmenyediakan failoverotomatis.UCARPadalah implementasiportabeldari protokolCARP.Implementasi Ucarp ini dapat dilakukan pada system operasi linux yang sejenis pada masing-masing server yang saling terhubung. Dibanding dengan konfigurasi menggunakan Heartbeat, konfigurasi UCARP lebih simple dan efisien

Server yang akan digunakan untuk UCARP masing-masing di konfigurasi memiliki IP virtual yang sama, namun salah satunya di set sebagai *master* dan yang satunya lagi di set sebagai *slave*. Konfigurasinya dapat dilihat pada gambar 7. Pada saat primary master server aktif, maka IP virtual UCARP hanya akan tampil di primary server, sedang pada secondary server

tidak tampak. IP yang diakses untuk kedua server ini adalah IP UCARP, yakni 192.168.1.233. Saat dilakukan ping ke IP 192.168.1.233, maka yang terdeteksi adalah IP dari primary server, yakni 192.168.1.101. tapi jika primary server di-off-kan, maka secara otomatis akan dialihakan ke secondary server dengan waktu pengalihan sekitar 2 detik lebih cepat beberapa detik dari *system failover* pada windows, hal ini dapat di lihat padatable 1.

Dari tabel1 terlihat ada peralihan dari primary server ke secondary server, dengan waktu peralihan 2 detik. Saat primary server kembali dinyalakan, maka secara otomatis UCARP akan mendeteksi primary server tersebut dan langsung mengalihkannya. Peralihan dari secondary server ke primary server membutuhkan waktu yang kurang lebih sama ketika terjadi peralihan. Dari hasil pengukuran pada cacti, rata-rata ketersediaan server dangan menggunakan teknik ini adalah 99.99 %.

Table 1. Data perbandingan failover cluster pada empat scenario

Skenario Pengukuran	Up Time (s)	Down Time (s)	Availability (%)
Server *stand alone*	16:25:00	7:34:00	68.403
Server *Failover Cluster Windows Server 2008 R2*	23:59:56	0:00:04	99.995
Server *Failover Cluster Proxmox VE*	23:59:58	0:00:02	99.998
Server *Failover Cluster Mesin Virtual Ubuntu Server 10.10*	23:59:59	0:00:01	99.999
Server *Failover Cluster Menggunakan DNS Failover*	23:59:58	0:00:02	99.998

Pengukuran Performance Virtual Virtual Ubuntu Server 10.10 Maveric Merkat

Pada table 1, failover cluster pada mesin virtual nilai *availability* mencapai 99,999%. Dari hasil pengukuran tersebut terlihat bahwa perpindahan dari mesin virtual yang satu ke mesin virtual yang lain sama sekali tidak memberikan efek yang berarti, tidak terlihat adanya respon perpindahan yang besar.Hal ini dikarenakan sistem pengukuran hanya menemukan nilai downtime yang sangat kecil pada server mesin virtual tersebut. Konsep *failover* yang digunakan sama dengan *failover* yang digunakan pada mesin Proxmox VE 2.1 yakni UCARP. Hasil yang diperoleh pada Proxmox VE 2.1 kurang lebih sama dengan hasil yang diperoleh pada mesin server virtualyang

nilai availabilitynya memenuhi standar yang ditetapkan sebesar 99,99%.

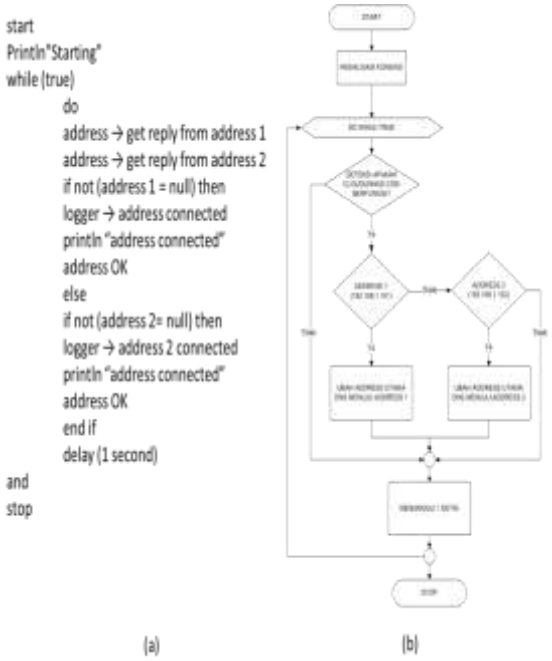
Hal ini disebabkan karena peralihan terjadi dalam satu mesin, dan tidak melibatkan mesin lain, sehingga perpidahan pada jaringan tidak menggunakan waktu yang lama. Selain faktor tersebut, saat perpindahan antar mesin fisik biasanya dibutuhkan load yang lama hingga mesin tersebut stabil dan bisa dicek kembali, sedang pada mesin virtual, sistemnya tidak demikian, load sistem lebih cepat.

Pengukuran Performance DNS Failover Pada Ubuntu Server 10.10

DNS *failover* adalah sistem failover yang bisa digunakan pada sistem operasi Linux dan Windows (Verryson, dkk. 2012). Prinsip kerja dari DNS-*failover* yaitu, master dan node diberi nama domain yang sama, dan nama inilah yang akan dipanggil. Jadi harus ada satu server yang berfungsi sebagai server DNS. Caranya yakni dengan membuat *file* dalam direktori */var/lib/bind.* Kemudian dilakukan perubahan pada file *named.conf.local* yang terletak dalam folder */etc/bind/*.

Selanjutnya kedua IP client pada jaringan local, diseting pada server DNS sehingga kedua IP tersebut berfungsi sebagai alamat dari *cloudunhas.com*.Semua client akan berhubungan dengan internet melalui server DNS. Agar konfigurasi ini dapat dikenali oleh mesin server, maka diperlukan restart pada bind, dengan perintah "*/etc/init.d/bind9 restart*".

Saat client mencari alamat dengan memasukkan nama domain, maka server DNS akan memberikan salah satu dari 2 alamat IP yang didaftarkan pada DNS server. Jika IP yang pertama gagal, maka akan dialihkan ke IP yang ke dua.Pada pengukuran DNS *Failover* digunakan script untuk melakukan looping perulangan ping. Jadi jika *primary* server mengalami gangguan, maka secara otomatis akan beralih ke IP *secondary*server, dan apabila *primary* server sudah normal kembali, maka akan dialihkan kembali ke alamat IP *primary* server. Script*check_network* yang mengatur agar proses ping dapat tetap berlangsung secara otomatis serta proses keja dari script tersebut dapat dilihat pada **gambar 4.**



Gambar 4.(a) *Script* check_network , (b) *Flowchart script* check_network DNS *Failover*

D. PEMBAHASAN

Pada **table 1,** terlihat bahwa server *stand alone* mengalami *gangguan* selama 7 jam, 34 menit, dengan waktu pengukuran 24 jam, sehingga waktu *uptime*-nya adalah 16 jam, 25 menit. Sedang pada server yang menerapkan sistem *failovercluster* pada *Windows* Server 2008 R2 mengalami *down time* selama 4 detik, sehingga waktu *uptime*-nya adalah 23 jam, 59 menit dan 56 detik. Demikian pula pada server yang menerapkan sistem *failovercluster* pada *Proxmox* mengalami *down time* selama 2 detik, sehingga waktu *uptime*-nya adalah 23 jam, 59 menit dan 58 detik. Pada *failover cluster* di mesin virtual *Ubuntu* Server 10.10mengalami *down time* selama 1 detik, sehingga waktu *uptime*-nya adalah 23 jam, 59 menit dan 59 detik. Sedang *failover cluster* yang menggunakan *DNS failover* mengalami *down time* selama 2 detik, sehingga waktu *uptime*-nya adalah 23 jam, 59 menit dan 58 detik. Dari data tersebut rata-rata persentasi *availability* failover cluster server pada empat scenario yang direncanakan, yakni pada Windows Server 2008 R2 *availabilitynya* sebesar 99.998%, pada Proxmox VE 2.1 *availabilitynya* mencapai 99.995%, pada mesin *virtual* Ubuntu Server 10.10 *availabilitynya* mencapai 99.999 % dan pada DNS failover, *avalabilitynya* 99.998%.Secara teoritis nilai availability tersebut diukur dengan persamaan $Availability = \frac{operation\ time - down\ time}{total\ time} \times 100\%$(Fatoni, 2010).

Tabel 2. Data Hasil Pengukuran *Workload* Server Untuk Masinghy–Masing Skenario

Skenario Pengukuran	Koneksi Per Detik
Server *stand alone*	190
Server *Failover Cluster Windows Server 2008 R2*	235
Server *Failover Cluster Proxmox VE*	251

Server *Failover Cluster Mesin Virtual Ubuntu Server 10.10*	253
Server *Failover Cluster Menggunakan DNS Failover*	252

Berdasarkan pada **tabel 2**, dapat disimpulkan bahwa server dengan sistem *failover cluster* lebih baik dibandingkan server yang *stand alone,* yang mana pada server *stand alone,* koneksi yang dicapai hanya 190 per detiknya, sedangkan server yang menggunakan sistem *failover cluster* bisa mencapai 253 koneksi per detiknya. Hal ini dapat terjadi karena pada sistem ada pembagian beban kerja, hingga saat koneksi terputus, ada sistem yang melakukan *backup.*

Tabel 3. Data Hasil Pengukuran *latency* Server Untuk Masing–Masing Skenario

Skenario Pengukuran	Latency (ms)
Server *stand alone*	50
Server *Failover Cluster Windows Server 2008 R2*	2
Server *Failover Cluster Proxmox VE*	0,3
Server *Failover Cluster Mesin Virtual Ubuntu Server 10.10*	5
Server *Failover Cluster Menggunakan DNS Failover*	2,41

Pada **tabel 3**, dapat terlihat bahwa sistem *failover cluster* maupun sistem yang *stand alone* mempunyai *latency* dipengaruhi oleh besarnya delay yang terjadi saat *client* menuju ke server. Karena pengujian dilakukan di jaringan lokal, maka *latency*nya kecil, yakni *hop* yang dilewati hanya satu.Untuk melihat jumlah *hop* yang dilewati, cukup dengan mengetik perintah *tracert* "alamat server yang dituju" pada *command promp*. Hanya saja saat melakukan gangguan pada server *stand alone*, terjadi pemutusan hubungan (*request time out*), yang mana *latency (delay*) saat server terkoneksi kembali dicatat sehingga *latency (delay*) yang tercatat hingga 50 ms. Berbeda dengan server yang diterapkan sistem *failover cluster*, walaupun server diberi gangguan, ada server backup sehingga tidak terjadi *delay* yang berarti. Pada sistem *failover* dengan *DNS Server,* saat perpindahan dari server utama ke server *backup* terjadi *delay* hingga 22,6 ms, sehingga *latency* mencapai 2,41 ms.Makin kecil *latency* maka kualitas jaringan semakin baik.

Tabel 4. Data Hasil Pengukuran *Paket Loss* Server Untuk Masing–Masing Skenario

Skenario Pengukuran	Packet Loss (%)
Server *stand alone*	7
Server *Failover Cluster Windows Server 2008 R2*	0
Server *Failover Cluster Proxmox VE*	3
Server *Failover Cluster Mesin*	0

Virtual Ubuntu Server 10.10	
Server *Failover Cluster Menggunakan DNS Failover*	0

Berdasarkan data pada **tabel 4**, *Packet loss* terbesar terjadi pada server *stand alone*, pada saat server diberi gangguan yang menyebabkan gagalnya fungsi server, maka paket data yang hilang sebesar 7%. Gangguan yang diberikan hanya sekali dengn waktu yang *relative* sangat singkat (lihat lampiran halaman L6 - 112).*Packet loss* adalah banyaknya paket data yang gagal mencapai server tujuan.Jika *packet loss* tinggi, kinerja jaringan buruk. Dalam kasus server *stand alone*, seperti yang dijelaskan pada bagian sebelumnya,server tidak memiliki backup, sehingga saat gagal, jaringan tidak dapat mendeteksi keberadaan server tersebut lagi. Berbeda pada sistem *failover cluster*, yang mana ID dari server yang terdeteksi pada pada jaringan akan terus ada, kecuali jika server utama dan server backup dalam kondisi *off* atau *down.Packet loss* pada server yang menggunakan sistem *failover cluster* sebagian besar adalah 0%, kecuali pada server yang menggunakan *Proxmox VE*, yang mana *packet loss*-nya mencapai 3%. Kondisi ini disebabkan oleh adanya kegagalan jaringan saat pengujian pada server *Proxmox VE.*

E. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan pada implementasi *failover cluster* pada *platform* yang berbeda, yang mana penelitian mengukur kinerja server dan jaringan, maka dapat ditarik kesimpulan, kinerja server yang menerapkan sistem *failover cluster* lebih baik dibandingkan dengan kinerja server *stand alone*, hal ini berdasarkan parameter *availability*, maksimum *workload web server*, nilai *latency*-nyaserta *paket loss* yang diperoleh.Sistem *failover cluster* menggunakan UCARP dan sistem yang menggunakan DNS *failover*sama baiknya dari segi kinerja, tapi dari segi pemakaian *DNS failover*sedikit lebih baik, karena bisa digunakan pada *platformLinux* dan *Windows*.Sistem *failovercluster* dengan menggunakan NAS tidak memindahkan aplikasi, tetapi hanya menangani pemindahan file. Untuk penelitian selanjutnya disarankan untuk mengantisipasi apabila terjadi *Fail*pada server, maka sebaiknya memadukan antara tiga teknik penaganan fungsi server yaitu *failover cluster*, *load balancing* dan miroring.

DAFTAR PUSTAKA

[1] Anonim. (2010). *Providing An innovative Solution To Address Planned And Unplanned Downtime For The Most Important Application.* USA.

[2] Bajgoric, N. (2009), *Continuous Computing Technologies for Enhancing Business Continuity,InformationScienceReference*,NewYork.

[3] Sahid, Ahmad Abdillah. *Penerapan Cluster Table Pada Basis Data perpustakaan Online dengan Oracle 11g*. Mercubuana, Jakarta

[4] Shimonski, Robert (2003), Windows 2003Clustering and Load Balancing. McGraw-Hill/Osborne.

[5] Dharma, Surya. (2008). *Pendekatan, Jenis dan Metode Penelitian Pendidikan*. Jakarta.

[6] Veryssoon, dkk. (2012), Simulasi Load Balancing Multihoming dan Fail-Over menggunakan VYATTA. Politeknik Telkom, Bandung.

[7] Bel, Laurent, 2012, *Simple Failover Cluster On Ubuntu Using CARP*.

[8] Fatoni.(2010), *Analisis Kualitas Layanan Jaringan Internet*. Palembang, Universitas Bina Darma.

[9] Garnieri, H. Megan.(2010), *Design and Implementation of Server Virtualization in Thiess Contractors Indonesia*, UGM, Yogyakarta.

[10] Hirt Alan. (2009). *Pro SQL Server 2008 Failover Clustering*, Apress, New York.

[11] Komaruddin (2012), *Implementasi Clustering Pada Jaringan Sistem Diskless Menggunakan Red Hat Enterprise Linux 5*. Politeknik Telkom, Bandung