Analisa Kinerja pada *Standalone Server* dan *Clustering Server* Teknologi RAC (*Real Application Clustering*) dengan Algoritma DNS (*Domain Name System*) Round Robin Berbasis Oracle Linux 6.4 di Lingkungan Virtual

M P Hapsari[1], Agung Budi Prasetijo[2], Adnan Fauzi[3]

Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudarto No.13, Tembalang, Kec. Tembalang, Kota Semarang, Jawa Tengah 50275

[1)] *mphapsari@student.ce.undip.ac.id*
[2)] *agungprasetijo@gmail.com*
[3)] *adnan@live.undip.ac.id*

ABSTRACT *–A standalone server is a server that runs applications and databases on the same computer or server. As a service provider, the technology on the server is required to continue to develop in line with the needs. Therefore, a server clustering architecture was developed to serve the needs of clients on a larger scale. One solution to this problem is by implementing the RAC (Real Application Clustering) technology provided by Oracle 11gR2. RAC technology can fulfill client needs by using Round Robin DNS (Domain Name Server) algorithm. Then, this technology can help add new servers and recently assigned servers as cluster members. Also, RAC has advantages of memory storage for database storage. Based on the results of tests that have been done, server clustering has better quality according to a standalone server. Testing is done using QOS (Quality of Service) testing which is carried out on the network and also on load tests using TPM (Transaction per Minute) which is carried out on the database.*

Keywords - Cluster and Standalone Server; DNS Round Robin; QOS testing; RAC Technology; TPM testing.

1. BAB I PENDAHULUAN

Server menyediakan berbagai macam layanan. Beberapa diantara layanan tersebut adalah otentikasi, keamanan, Web, berkas dan data. Dalam sebuah jaringan terdapat minimal satu buah server yang berfungsi untuk menyediakan layanan untuk klien [1]. Arsitektur server terbagi menjadi beberapa diantaranya *standalone server* dan *two tier*. *Standalone server* menjalankan aplikasi dan basis data pada komputer/server yang sama. Arsitektur tunggal awalnya digunakan untuk server mainframe seperti UNIX, AS400, dan sebagainya [2]. Seiring berkembangnya teknologi, kebutuhan akan fungsi server semakin meningkat, terutama pada dunia *enterprise.* Oleh karena itu fungsi *standalone server* dapat dikembangkan menjadi *clustering server*. *Clustering server* merupakan kumpulan dari beberapa server yang berdiri sendiri kemudian bekerja sama menjadi suatu sistem tunggal. Dengan clustering, basis data yang disimpan dapat terbagi ke beberapa mesin dan pada saat aplikasi berjalan semua mesin yang menyimpan data tersebut dianggap sebagai satu kesatuan.

Setiap server telah menentukan batas beban yang dapat ditanggung, sehingga setiap server memiliki batasan jumlah klien yang terhubung pada satu waktu. Kemampuan tersebut bergantung pada pengaturan basis data server itu sendiri, jenis permintaan HTTP (*Hypertext Transfer Protocol*), jenis konten, kondisi konten di server dilakukan cache atau tidak, juga perangkat keras, perangkat lunak serta sistem operasi yang digunakan. Ketidakmampuan server menanggung beban dapat menimbulkan macetnya layanan. Oleh karena itu dibutuhkan suatu mekanisme untuk membagi beban ke beberapa server, sehingga beban tidak berpusat di satu server saja. Kemampuan tersebut dikenal dengan nama *load balancing* [3]. Berikut terdapat beberapa penelitian terdahulu yang dijadikan acuan dalam membuat penelitian.

Penelitian pertama merupakan jurnal yang berjudul, "Analisis Algoritma Round Robin, Least Connection dan Ratio pada *Load Balancing* menggunakan OPNET Modeler" merupakan pengujian *load balancing* pada tiga algoritma yang berbeda menggunakan aplikasi OPNET Modeler 14.5 simulator. Penelitian ini berisi pengujian kinerja *load balancing* pada HTTP, FTP dan VoIP. Penelitian ini menunjukkan bahwa algoritma ratio memiliki kinerja yang lebih baik daripada algoritma least connection dan Round Robin. Ratio dapat menangani *end-to-end delay* dan *jitter* pada VoIP sedangkan least connection memiliki nilai *throughput* tertinggi di semua pengujian [4].

Penelitian kedua yang juga menjadi referensi

berasal dari jurnal yang berjudul "Analisis Performa *Load Balancing* DNS Round Robin dengan Linux *Virtual Server* pada Webserver Lokal" merupakan pengujian *load balancing* pada dua algoritma yang berbeda pada webserver local. Hasil penelitian ini menunjukkan bahwa algoritma *linux virtual server* memiliki availabilitas dan skalabilitas yang lebih baik daripada DNS Round Robin. *Linux virtual server* melayani *request* lebih baik dari DNS Round Robin, selain itu DNS Round Robin lemah dari sisi *packet loss* saat keadaan server *down* [5].

Penelitian ketiga berasal dari jurnal yang berjudul "Simulasi Pengclusteran Oracle 10G dengan *Real Application Clustering* (RAC)" merupakan perancangan *clustering server* dan *standalone server* menggunakan Oracle 10g di lingkungan virtual. Pengujian yang digunakan pada penelitian ini menggunakan model *Order Entry.* Hasil penelitian menunjukkan bahwa *standalone server* menghasilkan TPM (*Transaction per Minute*) lebih besar 1,6-1,8 kali lebih besar daripada *clustering server*, serta *standalone server* menghasilkan waktu *delay* lebih kecil daripada *clustering server*. Hal ini diakibatkan oleh peranan penggunaan dua *virtual machine* pada server basis data yang tercluster [6].

Penelitian keempat berasal dari Tugas Akhir yang berjudul "Analisis dan Perancangan Server Virtual pada Agiva Education" merupakan perancangan server untuk mengetahui perbandingan performa dan *cost* antara server virtual dan server fisik. Pengujian performa dilakukan menggunakan HammerDB dengan memberikan jumlah *virtual user* yang berbeda. Hasil penelitian menunjukkan bahwa server fisik dengan RAM 16 GB memiliki nilai TPM (*Transaction per Minute*) lebih besar daripada server virtual dengan RAM 4 GB [7].

Penelitian pertama dan kedua membandingkan kinerja antara sesama server cluster saja. Penelitian keempat membandingkan kinerja antara sesama server standalone saja. Hanya terdapat satu penelitian terdahulu yaitu penelitian ketiga yang membandingkan kinerja antara server cluster dengan server standalone. Namun, penelitian ketiga memiliki perbedaan pada penerapan versi pada Oracle dengan penelitian yang dilakukan Penulis. Versi Oracle pada penelitian terdahulu adalah Oracle 10g sedangkan pada penelitian ini menggunakan Oracle 11g Release 2. Kedua versi tersebut menerapkan Grid sebagai teknologi komputasi. Namun Oracle 11gR2 memiliki kelebihan pada fitur keamanan. Oracle 11gR2 memiliki otentikasi berbasis password dengan penggabungan *case* password, enkripsi tingkat *tablespace*, dan sebagainya. Berikut merupakan tinjauan pustaka yang dijadikan acuan melakuan penelitian.

Standalone server adalah server yang berjalan sendiri dan bukan merupakan bagian dari grup sehingga keamanan yang kompleks serta autentikasi tidak terlalu dibutuhkan [8]. *Clustering server* melibatkan satu grup server yang menerima trafik dan membagi tugas di antara grup server tersebut.

Load balancing atau pemerataan beban adalah sebuah metode dalam jaringan komputer yang dibutuhkan bila trafik data semakin meningkat. *Load balancing* menjadi solusi yang digunakan untuk mengatur masalah redudansi, skalabilitas dan manajemen yang dibutuhkan[9].

Dalam dunia jaringan komputer *load balancing* dapat diukur melalui performa atau QoS (*Quality of Service*). QoS merupakan kemampuan jaringan untuk menyediakan layanan yang lebih baik untuk trafik tertentu dari berbagai macam teknologi seperti jaringan IP, frame relay, dan sebagainya. Beberapa parameter yang dijadikan sebagai tolok ukur untuk mengukur QoS antara lain sebagai berikut [10].

a. *Throughput*

Throughput adalah jumlah bit yang diterima dengan sukses per detik melalui sebuah sistem atau media komunikasi.

$$Throughput = \frac{Jumlah\ paket\ yang\ diterima}{Total\ waktu\ pengiriman\ paket} \quad (1)$$

b. *Jitter*

Jitter adalah variansi *delay*, yaitu perbedaan selang waktu kedatangan antar paket di terminal tujuan. Table 2.1 menunjukkan berbagai kategori degradasi yang ada pada parameter *jitter* berdasarkan versi ITU-T G.114.

Tabel 2.2 Parameter *Jitter* berdasarkan ITU-T G.114

Nilai *Jitter*	Kualitas
0 – 20 ms	Baik
20 – 50 ms	Cukup
>50 ms	Buruk

Berikut merupakan rumus penghitungan *jitter*.

$$Jitter = \frac{Total\ Variansi\ Delay}{Total\ Paket\ yang\ Diterima-1} \quad (2)$$

Sedangkan rumus untuk mencari total variansi *delay* adalah sebagai berikut.

$$Total\ Variansi\ Delay = (delay\ 2 - delay\ 1) + (delay\ 3 - delay\ 2) + \ldots + \big(delay\ n - delay\ (n-1)\big) \quad (3)$$

c. *Delay*

Delay adalah waktu tunda paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuanya [11]. Tabel 2.2 menunjukkan berbagai kategori yang ada pada *latency* atau *delay* berdasarkan KPI standar ITU-T G.114

Tabel 2.3 Parameter *Delay* berdasarkan ITU-T

G.114

Nilai *Delay*	Kualitas
0 – 150 ms	Baik
150 – 400 ms	Cukup, masih dapat diterima
>400 ms	Buruk

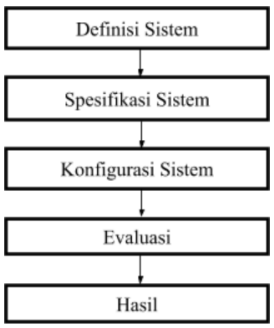
DNS Round Robin adalah algoritma yang membagi domain tunggal ke IP address yang berbeda. DNS merupakan sistem penamaan IP address pada perangkat komputer. Salah satu perangkat komputernya adalah server. Sehingga server yang memiliki IP address tersebut diakses melalui namanya saja. Bentuk implementasi dari DNS yang popular adalah BIND (*Berkeley Internet Name Domain*) [12].

RAC (*Real Application Clustering)* adalah teknologi yang dikembangkan oleh perusahaan oracle pada clustering server dengan mengedepankan fungsi ketersediaan tinggi dan skalabilitas. Basis data dengan menggunakan RAC dapat memungkinkan beberapa instance berjalan pada server yang berbeda untuk mengakses basis data yang disimpan pada penyimpanan bersama[13].

2. BAB II METODE DAN BAHAN

Diagram Blok

Metodologi penelitian yang digunakan untuk merancang bangun sistem server standalone dan server cluster di atas lingkungan virtual adalah DSRM (*Design Science Research Method)*, tahapan-tahapan dalam DSRM digambarkan pada Gambar 1 [14].



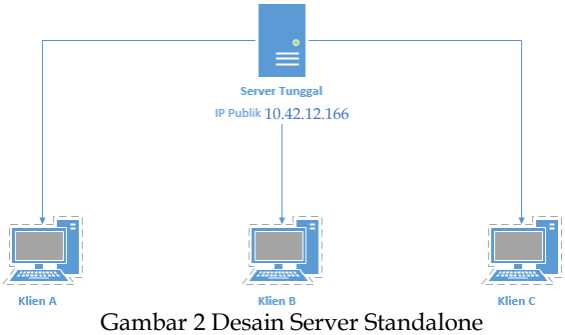
Gambar 1 Metode Penelitian DSRM (*Design Science Research Method*)

Aktifitas dari tiap tahapan dalam model ini adalah sebagai berikut:

a. Definisi Sistem
   Mendefinisikan sistem yang akan dibuat, meliputi penjabaran sistem, identifikasi kebutuhan sistem, tujuan dan manfaat sistem, cara kerja dan topologi yang digunakan.
b. Spesifikasi Sistem
   Proses spesifikasi kebutuhan akan menjabarkan tentang awal perancangan sistem dengan menentukan spesifikasi kebutuhan yang sesuai definisi sistem. Spesifikasi kebutuhan terdiri atas spesifikasi perangkat keras dan perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan.
c. Konfigurasi Sistem
   Pada tahap ini, spesifikasi kebutuhan yang telah ditentukan akan dirancang sesuai topologi/desain jaringan dan direalisasikan sebagai serangkaian sistem atau unit sistem yang memungkinkan untuk menjalankan tujuan sistem dan cara kerja sistem.
d. Evaluasi
   Proses evaluasi terdiri dari pengujian kinerja sistem dan menganalisis sistem secara keseluruhan untuk menjamin bahwa persyaratan sistem telah terpenuhi.
e. Hasil
   Penelitian dianggap berhasil jika sistem yang dibuat telah memenuhi tujuan dari penelitian.

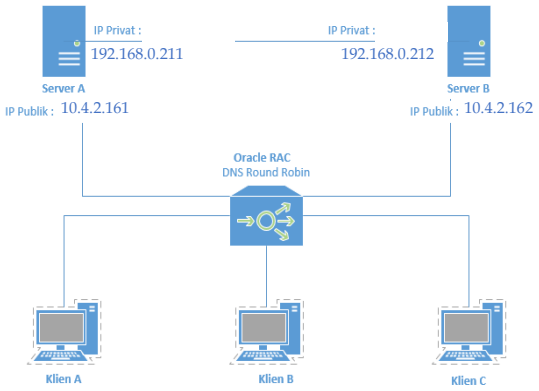
Topologi Jaringan

Desain topologi server cluster akan dibangun seperti pada Gambar 2.



Gambar 2 Desain Server Standalone

Gambar 2 merupakan server standalone dapat langsung terhubung dan melayani *request* dari beberapa klien tanpa menggunakan algoritma *load balancing*. Server standalone yang dibangun secara virtual dan tertanam di dalam sebuah komputer *host*. Server tersebut akan memiliki satu buah IP address yaitu IP Publik. Desain topologi server cluster akan dibangun seperti pada Gambar 3.



Gambar 3 Desain Server Cluster

Gambar 3 merupakan sistem cluster yang terdiri dari dua buah server standalone sebagai anggota cluster dan terhubung dengan sistem Oracle RAC dengan algoritma DNS round robin yang akan memproses permintaan dari klien. Sistem ini pun dibangun secara virtual di dalam sebuah komputer *host*. IP Publik dan IP Privat diimplementasikan menggunakan *network adapter* yang tersedia pada VirtualBox.

Kebutuhan Perangkat Keras

Kebutuhan perangkat keras pada penelitian digunakan untuk merancang spesifikasi yang berkaitan dengan komputer *host* serta server yang akan dibangun. Spesifikasi komputer *host* disesuaikan dengan Tabel 1.

Tabel 1. Spesifikasi Komputer *Host*

Komponen	Server
Prosesor	Intel Core i5-7200U CPU @ 2.50 GHz (4 CPUs)
Motherboard	X442UR 1.0
RAM	8192 MB
HDD	1 TB
NIC	2 *Fast Ethernet*

Berikut merupakan spesifikasi server standalone yang akan disesuaikan dengan Tabel 2.

Tabel 2. Spesifikasi Server Standalone

Komponen	Server
RAM	3072 MB
HDD	40 GB
NIC	1 *Fast Ethernet*

Berikut merupakan spesifikasi server cluster yang akan disesuaikan dengan Tabel 3.

Tabel 3. Spesifikasi Anggota Server Cluster

Komponen	Server I	Server II
RAM	3072 MB	3072 MB
HDD	40 GB	40 GB
NIC	2 *Fast Ethernet*	2 *Fast Ethernet*

Berikut merupakan spesifikasi HDD external disesuaikan dengan Tabel 4.

Tabel 4. Spesifikasi *Harddisk External* untuk Server Cluster

Komponen	Disk
HDD External	SATA 10 GB @ 5 buah

Kelima HDD ini akan dimasukan ke dalam diskgroup yang akan diatur oleh Oracle ASM (Automatic Storage Management) yang berfungsi untuk penyimpanan basis data pada kedua server cluster.

Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut.

1. Oracle Linux 6.4, digunakan sebagai sistem operasi untuk menjalankan server.
2. Oracle VM Virtualbox, digunakan untuk membuat server standalone dan server cluster secara virtual pada komputer *host*.
3. Oracle Linux 6.4 (x64), digunakan untuk sistem operasi pada server. Pemilihan Oracle Linux dikarenakan sistem operasi tersebut *free,* paket yang tersedia cukup lengkap, serta mendukung untuk instalasi *Oracle Database* dan *Oracle Grid Infrastructure*.
4. Oracle Database 11gR2 Grid Infrastucture for Linux (x64), dibutuhkan untuk instalasi Oracle ASM (*Automatic Storage Management*).
5. Oracle Database 11gR2 for Linux (x64), di dalamnya terdapat beberapa element seperti perangkat lunak oracle, basis data yang berisi file-file pada satu disk maupun lebih, *oracle instance*, pemrosesan server, *oracle net*.
6. Putty, digunakan untuk melakukan *remote access* ke mesin server. Putty adalah perangkat lunak *remote consol* yang digunakan untuk melakukan *remote access* ke komputer dan perangkat jaringan dengan menggunakan SSH, Telnet, atau serial.
7. Xming, digunakan untuk menyediakan tampilan X server pada Windows.
8. Siege, digunakan untuk pengujian *throughput*.
9. Wireshark, digunakan untuk pengujian *delay* dan *jitter*.
10. HammerDB, digunakan untuk pengujuan TPM (*Transaction Per Minute*) dengan menggunakan sejumlah klien virtual untuk memberi *request* pada sistem.

3. BAB III HASIL DAN PEMBAHASAN

Pengujian QoS dilakukan pada sisi klien. Parameter yang digunakan untuk pengujian ini adalah *packet loss, throughput, concurrency, delay,* dan *jitter*. Sedangkan pengujian dilakukan menggunakan terminal dan aplikasi wireshark dari sisi klien. Berikutnya adalah pengujian beban pada basis data, parameter yang digunakan untuk pengujian ini

adalah TPM (*Transaction per Minute*). Pengujian beban menggunakan aplikasi hammerdb.

Pengujian QOS (*Quality of Service*)

a. Pengujian *Packet Loss*

Dengan menggunakan pengujian *ping,* maka dapat diketahui jumlah *packet loss* yang terjadi pada setiap server. Presentase perubahan dari pengujian *ping* akan dijelaskan pada Tabel 5.

Tabel 5. Hasil Perbandingan *Packet Loss* antara Server Standalone dan Server Cluster

No.	Percobaan	*Packet Loss* (%)		Persentase perubahan (%)**
		Standalone	Cluster	
1.	Percobaan 1	0	0	0 %
2.	Percobaan 2	0	0	0 %
3.	Percobaan 3	0	0	0 %
4.	Percobaan 4	0	0	0 %

** = semakin kecil semakin baik

Berdasarkan Tabel 5 dapat dilihat jumlah *packet loss* bernilai 0% dan masuk dalam kategori sangat baik (0%) [15]. Sehingga tidak terjadi perunahan kualitas antara server standalone dan server cluster.

b. Pengujian *Throughput*

Pengujian *throughput* dilakukan dengan menggunakan perintah siege. Perintah siege dapat mensimulasikan sejumlah permintaan HTTP dengan sejumlah klien tertentu dan dalam waktu tertentu. Tabel 6 akan menunjukkan persentase perubahan nilai *throughput*.

Tabel 6. Hasil Perbandingan *Throughput* antara Server Standalone dan Server Cluster

No.	Percobaan	*Throughput (MB/sec)**		Persentase perubahan (%)*
		Standalone	Cluster (Rerata)	
1.	25 klien	0.05	0.05	0%
2.	50 klien	0.10	0.10	0%
3.	75 klien	0.15	0.15	0%
4.	100 klien	0.20	0.20	0%

* = semakin besar semakin baik

Dari Tabel 6 dapat disimpulkan bahwa nilai *throughput* tidak mengalami perubahan kualitas pada server cluster terhadap server standalone. Perubahan sebesar 0% dialami oleh kedua server pada kategori pengujian 25 klien, 50 klien, 75 klien dan 100 klien.

c. Pengujian *Concurrency*

Pengujian *concurrency* dilakukan dengan menggunakan perintah siege. Tabel 7 akan menunjukkan persentase perubahan nilai *concurrency*.

Tabel 7. Hasil Perbandingan *Concurrency* antara Server Standalone dan Server Cluster

No.	Percobaan	*Concurrency**		Persentase perubahan (%)*
		Standalone	Cluster (Rerata)	
1.	25 klien	7.84	7.92	1.02%
2.	50 klien	15.85	16.04	1.19%
3.	75 klien	23.81	23.83	0.08%
4.	100 klien	31.85	31.65	-0.6%

* = semakin besar semakin baik

Dari Tabel 7 dapat disimpulkan bahwa nilai *concurrency* mengalami peningkatan kualitas pada 25 klien sebesar 1.02%, 50 klien sebesar 1.19% dan pada 75 klien sebesar 0.08%. Namun mengalami penurunan kualitas pada 100 klien sebesar 0.6%.

d. Pengujian *Delay*

Skenario dalam melakukan pengecekan perintah *delay*, dilakukan dengan cara *ping* ke 10.42.12.166 pada server standalone, *ping* ke rac.cluster.com pada server cluster sebanyak 10 paket dan *delay* dilihat menggunakan aplikasi *wireshark*, sehingga didapatkan data untuk mengetahui *delay* dalam paket tersebut. Tabel 8 akan menunjukkan persentase perubahan nilai *delay*.

Tabel 8. Perubahan Persentase *Delay* Server Standalone dan Server Cluster

No.	Percobaan	*Delay* (s)		Persentase perubahan (%)**
		Standalone	Cluster	
1.	Percobaan 1	0.002530	0.000402	-84.1%
2.	Percobaan 2	0.001319	0.000518	-60.7%
3.	Percobaan 3	0.001225	0.001203	-1.7%
4.	Percobaan 4	0.001258	0.000318	-74.7%

** = semakin kecil semakin baik

Berdasarkan pada Tabel 8 disimpulkan kualitas *delay* dari server cluster mengalami peningkatan terhadap server standalone. Pada percobaan 1 nilai *delay* yang turun sebesar 84.1%, percobaan 2 nilai *delay* yang turun sebesar 60.7%, pada percobaan 3 nilai *delay* yang turun sebesar 1.7 % dan pada percobaan 4 nilai *delay* yang turun sebesar 74.7%. Namun hasil *delay* dari semuanya ini masuk dalam kategori baik yaitu dibawah 150ms [15].

e. Pengujian *Jitter*

Nilai pengujian *jitter* didapatkan dari penghitungan variansi *delay*. Tabel 9 akan menunjukkan persentase perubahan nilai *jitter*.

Tabel 9. Perubahan Persentase *Jitter* Server Standalone dan Server Cluster

No.	Percobaan	*Jitter*		Persentase perubahan (%)**
		Standalone	Cluster	
1.	Percobaan 1	0.003190	0.000290	-90.9%
2.	Percobaan 2	0.000653	0.000537	-17.7%
3.	Percobaan 3	0.000598	0.002112	258%
4.	Percobaan 4	0.000202	0.000207	2%

** = semakin kecil semakin baik

Berdasarkan hasil pada Tabel 9 terlihat dilihat bahwa pada server standalone cenderung mengalami penurunan nilai *jitter*, sedangkan pada server cluster nilai *jitter* cenderung fluktuatif, terutama pada Percobaan 3. Hal ini disebabkan karena nilai *delay* pada paket ke-7 memiliki rentang nilai yang cukup jauh dengan *delay* paket ke-6 dan *delay* paket ke-8, sehingga pada perhitungan menghasilkan nilai variansi yang besar. Faktor yang mempengaruhi *delay* diantaranya adalah jarak, transisi, transmisi ulang dan kapasitas jaringan. Namun, jarak dan kapasitas jaringan memiliki kondisi yang sama di setiap percobaan server. Sehingga transisi dan transmisi ulang dapat menjadi penyebab *delay* lebih besar dikarenakan server standalone dapat langsung terhubung dengan jaringan Laboratorium Jaringan Komputer, namun server cluster harus melalui perangkat Mikrotik untuk terhubung dengan jaringan tersebut.Namun, berdasarkan Tabel 2.1 Parameter *Jitter* berdasarkan ITU-T G.114, seluruh nilai *jitter* pada setiap percobaan memiliki kualitas baik karena berada pada rentang 0 – 20 ms. Jika dilihat pada Tabel 8 dan Tabel 9 nilai dari *jitter* masih dalam kategori baik dengan nilai kurang dari 20 ms [15].

Pengujian Beban / *Load Test*

a. Pengujian TPM (*Transaction per Minute*)

Berbeda dari ketiga pengujian sebelumnya yang digunakan untuk menguji QoS (*Quality of Service*) antara server dan klien, pengujian HammerDB digunakan untuk melakukan *benchmark test* dengan OLTP (*Online Transaction Processing*) dan menggunakan standar industri *benchmark* TPC-C (*Transaction Performance Processing Council*). Pengujian menggunakan satuan TPM (*Transaction per Minute*). OLTP adalah beban kerja yang diidentifikasi oleh basis data yang menerima permintaan data dan beberapa perubahan data dari sejumlah pengguna dari waktu ke waktu. Istilah modifikasi ini disebut juga transaksi.

Pada masing-masing server terdapat beberapa parameter yang ditetapkan, yaitu:

- Number of Warehouse: 33
- Total Transaction per Users: 1000000
- User Delay: 500 ms
- Repeat Delay: 500ms
- Iterations: 1

Tabel 10 akan menunjukkan persentase perubahan nilai TPM.

Tabel 10. Perubahan Persentase *Stress Test* pada Server Standalone dan Cluster

No.	Virtual User	*Transaction Counter*		Persentase perubahan (%)*
		Standalone	Cluster	
1.	5	282	390	38.2%
2.	10	486	558	14.8%
3.	15	384	468	21.8%

* = semakin besar semakin baik

Dari Tabel 10 dapat diketahui persentase perubahan dari transaksi server standalone terhadap server cluster mengalami peningkatan kualitas di setiap percobaan. Hal ini menanndakan bahwa server cluster mampu menangani beban lebih baik dari server standalone.

4. BAB IV KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah bahwa antara server standalone dan cluster memiliki kualitas yang sama saat pengujian *packet loss* dan pengujian *throughput* yaitu sebesar 0%. Server cluster memiliki keunggulan kualitas dari server standalone pada saat pengujian *delay* dan TPM (*Transaction per Minute*) pada keseluruhan percobaan. Dan terdapat ketidakstabilan pada pengujian *concurrency* dan *delay* di beberapa percobaan.

DAFTAR PUSTAKA

[1] R. M. Roberts, and C. William, "*Networking Fundamental*", Vol. 9, 2018, Pp 371-402.

[2] D. Lestari, "Sistem Penjualan Berbasis Client Server pada Bujel.net Kediri", AMIKOM Yogyakarta, Yogyakarta, 2013.

[3] H. Silalahi, "Pengertian Load Limit (Batas Beban)," Maret 2018. [Online]. https://andika-silalahi.blogspot.com/2012/07/pengertian-load-limit-batas-beban.html.

[4] H. Nasser, and T. Witono, "Analisis Algoritma Round Robin, Least Connection dan Ratio pada Load Balancing menggunakan OPNET

Modeler", Universitas Kristen Marathana, Bandung, 2016.

[5] A. J. Pradana, "Analisis Performa Load Balancing DNS Round Robin dengan Linux Virtual Server pada Webserver Lokal", Universitas Dian Nuswantoro, Semarang.

[6] W. Yunanto, and A. Wibowo, "Simulasi Pengclusteran Oracle 10G dengan Real Application Clustering (RAC)", Politeknik Caltex Riau, Riau.

[7] P. Virginia, F. P. Mulyono, H. Zufendi, and R. Tjiptadi, "Analisis dan Perancangan Server Virtual pada Agiva Education", Universitas Binus, Jakarta, 2013.

[8] --, "*Standalone server,*" Maret 2018. [Online]. https://www.techopedia.com/definition/13333/standalone-server.

[9] T. Bourke, "*Server Load Balancing*", Vol. I, 2001, pp 5-6.

[10] S. Haryadi, "*Quality of Service (QoS) dan Pengukurannya*", 2010.

[11] E. B. Setiawan, *Analisa Quality Of Services (Qos) Voice Over Internet Protocol (VOIP) Dengan Protokol H . 323 Dan Session Initial Protocol (SIP)*, Vol. 1, no. 2, 2012, pp. 1–8. [Ebook] Available: Komputa.

[12] C. Liu, and P. Albitz, "The History of BIND" in *DNS and BIND*, vol. 5, 2006, pp 9.

[13] Setiawan, I., "*Konsep dan Gambaran DNS/BIND*", http://pl.duniasemu.org/network/bind_dns/bind_dns-1.html, Maret 2018. (WEB)

[14] Alturki, A., Gable, G. G., & Bandara, W. *A design science research roadmap. In International Conference on Design Science Research in Information Systems*, 2011, pp. 107–123. [Ebook] Available: Springer.

[15] E.B. Setiawan, "Analisa Quality Of Services (QoS) Voice Over Internet Protocol (VoIP) Dengan Protokol H.323 Dan Session Initial Protocol (SIP)", UNIKOM, Bandung, 2012.