

FAILOVER CLUSTER WEB SERVER DATABASE PADA PROXMOX

Rudi Gunawan¹, Merry Agustina², Aan Restu Mukti³

^{1,2,3}Universitas Bina Darma

^{1,2,3}JalanJenderal Ahmad Yani No. 3 Palembang

rudigunawan133@gmail.com¹, merry_agst@binadarma.ac.id², aanrestu@binadarma.ac.id³

ABSTRACT

Systems with high availability are now needed to access information anywhere and anytime. For this reason, a system that can provide high availability is needed. Failover clusters are a group of servers that work together to maintain the availability of applications and services at a high level, if one server or node fails, another node will take over the workload without downtime. System clustering testing is done by testing unplanned failover cluster tests. From the results of testing with the unplanned failover test, it can be proven that if one of the nodes dies or a failure occurs then another node will take over the work automatically and there will also be a hard drive transfer from the master server to the backup server, then measure benchmark parameters on the node-1 server before and after the failover cluster system to compare better performance, the results after the cluster failover system were better because of the addition of a backup server.

Keywords : Failover Cluster, High Availability, unplanned failover test, benchmark

ABSTRAK

Sistem dengan tingkat ketersediaan tinggi sekarang ini dibutuhkan untuk mengakses informasi dimana saja dan kapan saja. Untuk itu diperlukan sistem yang dapat menyediakan *high availability*. *Failover cluster* merupakan sekelompok *server* yang bekerja sama dalam mempertahankan ketersediaan aplikasi dan layanan dalam tingkat yang tinggi, jika salah satu *server* atau *node* mengalami kegagalan maka *node* lain akan mengambil alih beban kerjanya tanpa *downtime*. Pengujian *system clustering* dilakukan dengan pengujian *unplanned failover cluster test*. Dari hasil pengujian dengan teknik *unplanned failover test* dapat dibuktikan bahwa jika salah satu dari *node* mati atau terjadi kegagalan maka *node* lain akan mengambil alih kerja secara otomatis dan terjadi juga perpindahan hardisk dari server master ke server backup, lalu mengukur parameter *benchmark* pada server node-1 sebelum dan sesudah adanya sistem *failover cluster* untuk membandingkan performa yang lebih baik, hasilnya sesudah adanya sistem failover cluster lebih baik karena adanya penambahan server backup.

Kata Kunci : Failover Cluster, High Availability, unplanned failover test, benchmark

1. PENDAHULUAN

1.1 Latar Belakang

Setiap instansi pemerintah maupun perusahaan selalu mengganti *server* lama menjadi *server* baru untuk kebutuhan aplikasi baru yang akan dijalankan. Lama-kelamaan pusat data menjadi penuh dengan *server* yang hanya digunakan sebagian kapasitas dari total kapasitas yang disediakan.

Virtualisasi adalah suatu cara untuk mengelola suatu *Data Center*. Virtualisasi pada *Cloud Computing* bisa dimanfaatkan untuk membangun beberapa *server virtual* di satu *server* fisik. Dengan begitu Perusahaan dapat menjalankan beberapa Sistem Operasi dan aplikasi di dalam satu perangkat dan tidak perlu membeli *server* baru ketika ada aplikasi lain yang ingin dijalankan. Selain itu setiap aplikasi yang dijalankan harus bisa diakses kapan saja dan dimana saja, karena itu perlunya sistem dengan tingkat ketersediaan yang tinggi atau dikenal dengan *High Availability*.

Proxmox adalah virtualisasi berbasis *hypervisor* atau yang dikenal juga dengan *virtual machine management service* (VMMs) yang memungkinkan beberapa *server virtual* dijalankan menggunakan satu *server* fisik. Selain itu *Proxmox* juga menyediakan *Failover Cluster*, *Failover Cluster* merupakan

sekelompok *server* yang bekerja sama dalam mempertahankan ketersediaan aplikasi dan layanan yang dalam tingkat yang tinggi.

Laboratorium komputer Universitas Bina Darma Palembang memiliki *server* yang berhubungan dengan *High Availability user* atau penyediaan data tingkat tinggi kepada pengguna. Namun hanya saja *server* yang telah disediakan oleh Laboratorium komputer Universitas Bina Darma untuk mengoptimalkan *resources hardware* belum sepenuhnya maksimal. Dalam prakteknya *server* itu ketika diakses sering terjadi kegagalan. Kegagalan itu sendiri diakibatkan karena *server down* dan tidak ada *system backup* dari sisi *server* lain untuk menggantikan posisi *server* dalam kondisi bermasalah. Hal ini memberikan dampak pada konektivitas layanan yang sedang berjalan. Sehingga Laboratorium komputer Universitas Bina Darma membutuhkan *system backup* dari sisi *server* lain untuk memecahkan masalah ini.

Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang didapat dalam penelitian *Failover Cluster Web Server Database* pada *Proxmox* adalah “Bagaimana *Simulasi Failover Cluster* pada *virtual web server database* menggunakan *Proxmox*?”.

Batasan Masalah

Untuk lebih terarah dan tidak menyimpang dari pokok masalah yang ada. Maka penulis membatasi pada permasalahan berikut ini:

Menggunakan *server 64-bit* yang mampu mendukung teknologi virtualisasi perangkat keras seperti *Intel Virtualization Technology (Intel-VT atau AMD-V)*.

1.1 Tujuan Penelitian

Tujuan yang akan dicapai dalam penelitian ini adalah sebagai berikut:

- a. Penelitian ini di Laboratorium Univeritas Bina Darma Palembang
- b. Mensimulasikan suatu sistem *failover cluster* yang menyediakan *high availability* dengan memanfaatkan teknologi *virtualisasi*
- c. Mengevaluasi performa *failover cluster* dengan mengukur parameter *benchmark*.

1.2 Waktu Penelitian

Tempat dalam melakukan penelitian tugas akhir ini yaitu berlokasi di Universitas Bina Darma Palembang dimulai dari Agustus 2018 sampai dengan Oktober 2018.

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Pengumpulan data merupakan tahap penting dalam penulisan karya ilmiah. Pengumpulan data menurut Sugiyono (2012:137) dapat dilakukan dengan berbagai *setting*, berbagai sumber, dan berbagai cara dalam upaya mengumpulkan data.

Adapun metode penelitian yang digunakan dalam penelitian ini ialah:

- a. Metode Wawancara (*interview*)

Dalam penelitian ini mengadakan wawancara atau tanya jawab langsung kepada kepala Laboratorium Universitas Bina Darma Palembang tentang keadaan server saat terjadi down atau kegagalan.

- b. Metode Studi Pustaka

Dengan melakukan pencarian dan pengumpulan buku-buku maupun informasi dari jurnal sebagai bahan referensi serta mempelajari dan memahami proses sistem failover cluster untuk melakukan prediksi nantinya.

2.2 Metode Penelitian

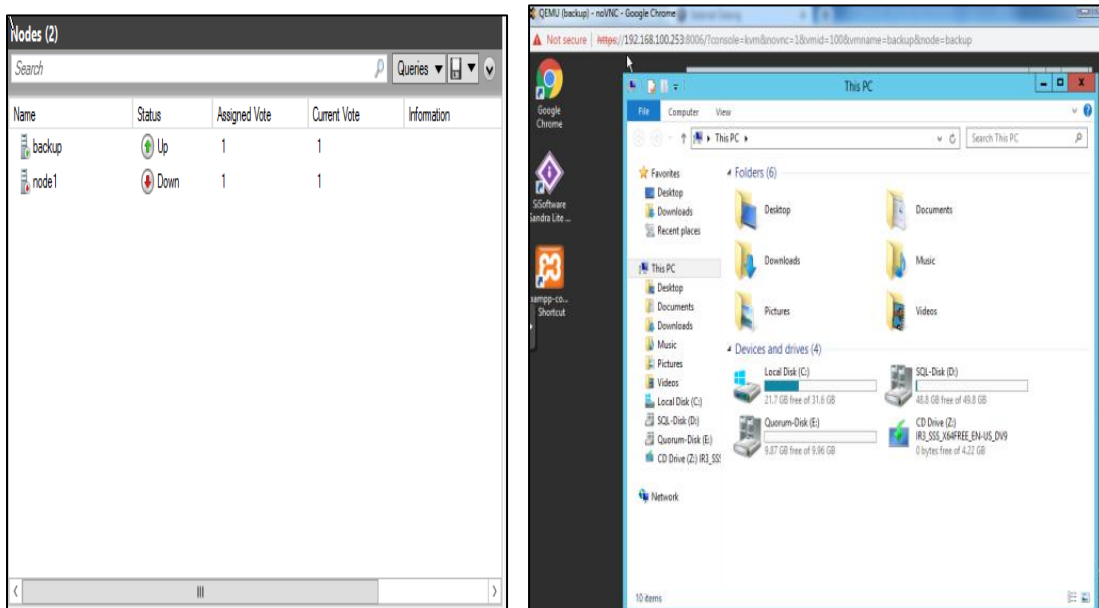
Metode yang digunakan pada penelitian ini yaitu Penelitian Tindakan (*action research*). Penelitian tindakan (*action research*) dilaksanakan bersama-sama paling sedikit dua orang yaitu antara peneliti dan partisipan atau klien yang berasal dari akademisi maupun masyarakat. Oleh karena itu, tujuan yang akan dicapai dari suatu penelitian tindakan (*action research*) akan dicapai dan berakhir tidak hanya pada situasi organisatoris tertentu, melainkan terus dikembangkan berupa aplikasi atau teori kemudian hasilnya akan di publikasikan ke masyarakat dengan tujuan riset. (Madya, 2006)

3. Hasil & Pembahasan

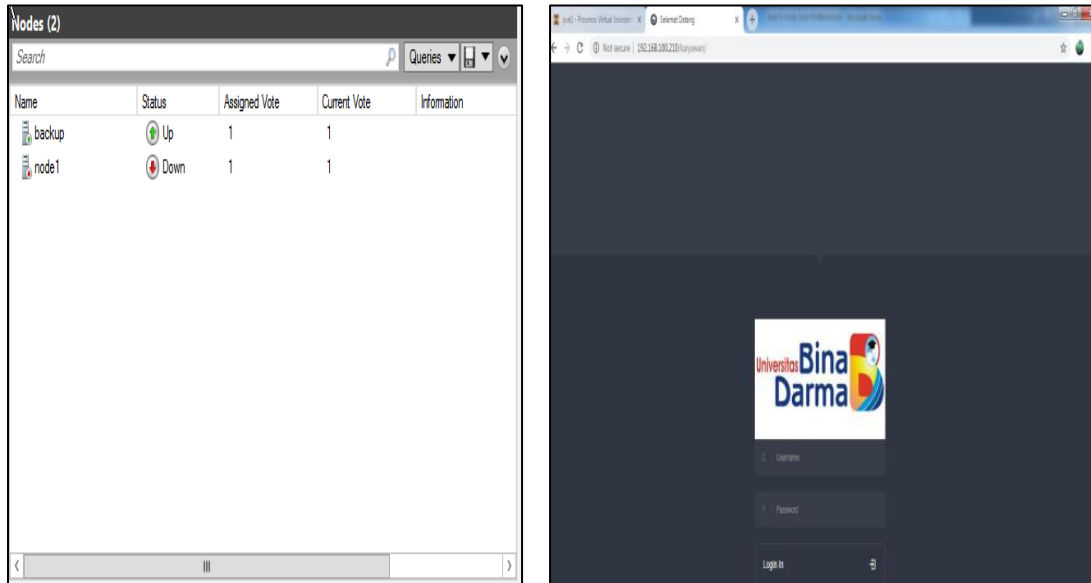
Hasil yang di dapat pada penelitian ini adalah sistem *failover cluster* yang dibuat dari sistem operasi *Windows Server 2012 R2 Data Center* yang diterapkan di mesin *virtual proxmox* kemudian dilakukan pengujian pada sistem *failover cluster* dengan cara pengujian *unplanned failover test* dan melakukan evaluasi sebelum dan sesudah adanya sistem *failover cluster*.

3.1 Hasil Pengujian

Adapun hasil penelitian ini yaitu peneliti akan melakukan pengujian dengan cara *unplanned failover test* yaitu pengujian terhadap *cluster* dengan cara mematikan salah satu *nodes cluster* kemudian dilihat apakah *nodes* lain dapat mengambil alih kerja *nodes* tersebut secara otomatis.



Gambar 1. Access Web saat satu *node* dalam keadaan mati dan terjadi Perpindahan iSCSI Disk



Gambar 2. Access Web saat satu node dalam keadaan mati

Pada gambar 1 dapat dilihat *iSCSI Disk* yang ada pada server *NODE-01* beralih pada server *back up* saat dimana server *NODE-01* dinonaktifkan. Pada gambar 2, peneliti mencoba untuk mengakses web server dan berhasil. Dari hasil pengujian dengan teknik *unplanned failover test* dapat dilihat bahwa jika salah satu dari *node* mati atau terjadi kegagalan maka *node* lain akan mengambil alih kerja secara otomatis.

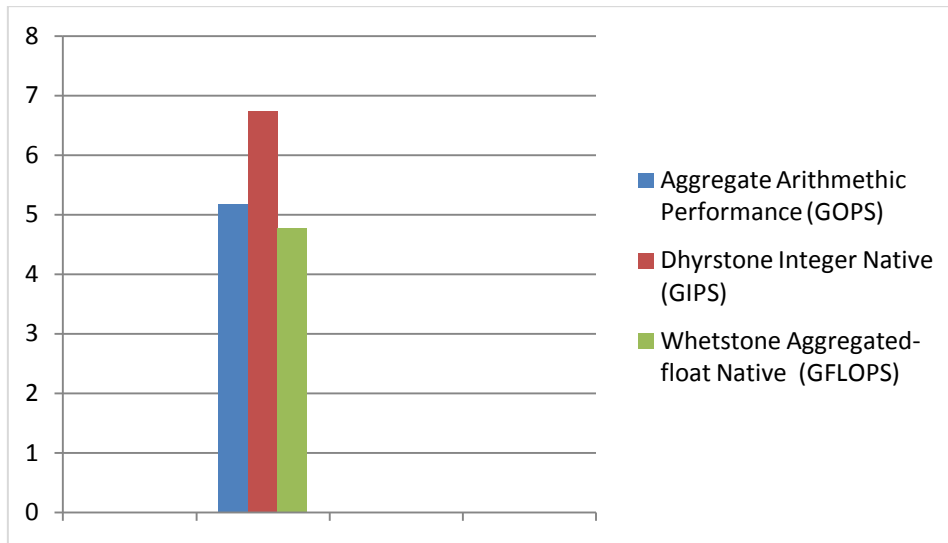
3.2 Hasil perbandingan

a. Processor Arithmetic

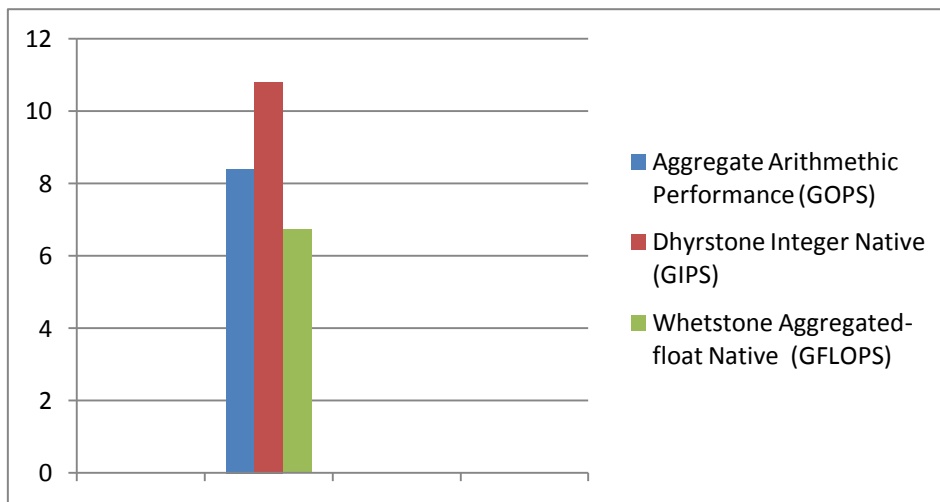
Untuk hasil yang didapat dari pengamatan untuk parameter *processor arithmetic* dapat dilihat pada tabel 1.

Tabel 1. Hasil Benchmark Processor Arithmetic

Tabel Hasil Benchmark Processor Arithmetic						
	Sebelum ada sistem failover cluster			Sesudah ada sistem failover cluster		
Uji Coba Ke-	Aggregate Arithmetic Performance (GOPS)	Dhystone Integer Native (GIPS)	Whetstone Aggregated-float Native (GFLOPS)	Aggregate Arithmetic Performance (GOPS)	Dhystone Integer Native (GIPS)	Whetstone Aggregated-float Native (GFLOPS)
1	5.09	6.61	4.7	8.09	10.41	6.61
2	5.2	6.78	4.8	8.45	10.87	6.78
3	5.21	6.77	4.83	8.48	11	6.77
4	5.2	6.78	4.81	8.5	11	6.78
5	5.2	6.75	4.8	8.52	11	6.75
Rata	5.18	6.73	4.78	8.40	10.85	6.73



Gambar 3. Grafik Hasil *Benchmark Processor Arithmetic* sebelum ada sistem *failover cluster*



Gambar 3. Grafik Hasil *Benchmark Processor Arithmetic* sesudah ada sistem *failover cluster*

Berdasarkan hasil pengamatan pada parameter *processor arithmetic* yang dapat dilihat pada Table 1 didapatkan bahwa sebelum adanya sistem *failover cluster node-1 / server master* memiliki nilai rata – rata yang lebih kecil dibandingkan sesudah adanya sistem *failover cluster* dan pada gambar 3. grafik bisa dilihat sesudah ada sistem *failover cluster* grafiknya meningkat dibandingkan gambar 2. grafik sebelum ada sistem *failover cluster* grafiknya menurun.

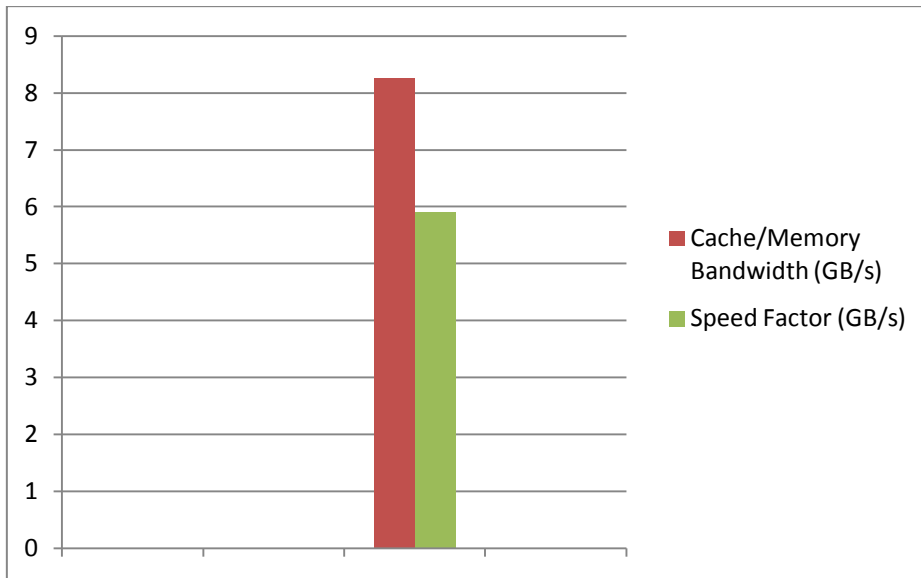
Dimana *Node-1 / server masters* sebelum adanya sistem *failover cluster* pada pengukuran *Aggregate Arithmetic Performance (GOPS)* memiliki rata-rata nilai 5.18 *GOPS*, pada pengukuran *Dhyrstone Integer Native (GIPS)* memiliki rata-rata nilai 6.73 *GIPS*, dan pada pengukuran *Whetstone Aggregated-float Native (GFLOPS)* memiliki rata-rata nilai 4.78 *GFLOPS* sedangkan sesudah adanya sistem *failover cluster* pada pengukuran *Aggregate Arithmetic Performance (GOPS)* memiliki nilai rata-rata 8.40 *GOPS*, pada pengukuran *Dhyrstone Integer Native (GIPS)* memiliki nilai rata-rata 10.85 *GIPS*, dan pada pengukuran *Whetstone Aggregated-float Native (GFLOPS)* memiliki nilai rata-rata 7.14 *GFLOPS*. Dari ketiga *instrument* pengukuran yang dilakukan dapat disimpulkan bahwa sesudah adanya sistem *failover cluster* performa menjadi lebih baik dibandingkan dengan sebelum adanya sistem *failover cluster* dikarenakan adanya penambahan sistem backup.

b. Cache and Memory

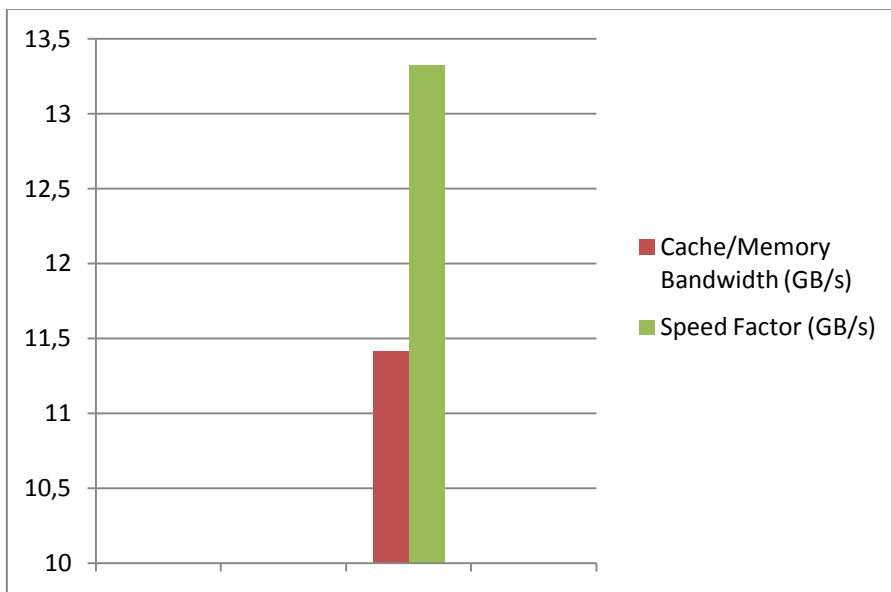
Parameter berikutnya yang akan dilakukan pengukuran yaitu *cache and memory*. Pada parameter ini akan ada dua kategori yang dilihat yaitu *bandwidth cache and memory* dan *speed factor*, untuk kategori *bandwidth cache and memory* dilihat berapa *bandwidth* yang didapat pada *memory* dimana untuk kategori *cache and memory* nilai pengukuran yang lebih besar menunjukkan performa yang lebih baik, sedangkan untuk kategori *speed factor* melihat kecepatan disaat melakukan akses terhadap komputer dimana untuk kategori *speed factor* nilai yang lebih besar menunjukkan performa yang lebih baik.

Tabel 2. Hasil Benchmark Cache and Memory

Tabel Hasil Benchmark Cache and Memory				
	Sebelum ada sistem <i>failover cluster</i>		Sesudah ada sistem <i>failover cluster</i>	
Uji Coba Ke-	Cache/Memory Bandwidth (GB/s)	Speed Factor (GB/s)	Cache/Memory Bandwidth (GB/s)	Speed Factor (GB/s)
1	8.36	5.70	10.036	15.3
2	8.2	6.20	9.91	14.2
3	8.37	5.80	12.17	12.60
4	8.18	5.70	12.2	13.10
5	8.15	6.10	12.76	11.40
Rata-rata	8.25	5.9	11.41	13.32



Gambar 4. Grafik Hasil *Benchmark Cache and Memory* sebelum ada sistem *failover cluster*



Gambar 5 . Grafik Hasil *Benchmark Cache and Memory* sesudah ada sistem *failover cluster*

Pada table 2. diatas terlihat nilai rata-rata untuk kategori *cache and memory* sebelum adanya sistem *failover cluster* berkisar 8.25 GB/s, sedangkan sesudah adanya sistem *failover cluster* berkisar 11.41 GB/s dan untuk *speed factor* sebelum adanya sistem *failover cluster* berkisar 5.9 GB/s/ sedangkan sesudah adanya sistem *failover cluster* berkisar 13.32 GB/ sedangkan pada grafik diatas menjelaskan pada gambar 5. grafik nilai rata-rata untuk kategori *cache and memory* sesudah adanya sistem *failover cluster* meningkat dibandingkan gambar grafik 4. sebelum adanya sistem *failover cluster*. Untuk grafik 5. nilai rata-rata *speed factor* sesudah adanya sistem *failover cluster* meningkat dibandingkan grafik 4. sebelum adanya sistem *failover cluster*.

4. KESIMPULAN

Akhirnya sudah sampai pada akhir dari laporan. Kesimpulan merupakan rangkuman singkat tentang pembahasan dari seluruh bab yang telah dibahas sebelumnya. Adapun kesimpulan pada penelitian yang berjudul “*Failover Cluster Web Server Database Pada Proxmox*” ini yaitu :

1. Dengan adanya *system Failover Cluster* diharapkan mampu menekan tingkat kegagalan pada *server* dan memudahkan untuk melakukan *maintenance* pada *server* seperti *backup* data dan lain-lain.
2. Dengan menggunakan *system clustering* dapat mempermudah *administrator* dalam hal manajemen *server* seperti saat *server primary* dalam keadaan mati atau sedang dilakukan *maintenance* maka secara otomatis beban kerja akan dialihkan ke *server secondary*.
3. Dari hasil test performa untuk dua parameter yaitu , *processor arithmetic*, dan *cache and memory*, masing–masing 8.40 GOPS, 5.18 GOPS , dan 11.41 GB/s , 8.25 GB/s.
4. Pada pengukuran *Benchmark Processor Arithmetic* yang dilakukan dapat disimpulkan bahwa sesudah adanya sistem *failover cluster* memiliki performa yang lebih baik dibandingkan sebelum sistem *failover cluster*.

DAFTAR PUSTAKA.

- [1] Afif (2013). Implementasi *Disaster Recovery Plan* Dengan Sistem *Fail Over* Menggunakan *DRBD* dan *Heartbeat* pada *Data Center FKIP UNS*.
- [2] Everest, G. C. (2005). *Fundamentals of Database System*. Tokyo: Benyamin.
- [3] Febriani. (2011). Implementasi *Server Cluster High Availability* pada *Web Server* dengan Sistem *Turnkey Linux* Menggunakan *Heartbeat*.
- [4] Kurniawan, E. R. S. I. F. (2016). Simulasi Sistem *Failover* Komputer *Clustering* Menggunakan *Hyper-v* pada *Windows Server 2012 R2*.
- [5] Lukitasari (2010). ANALISIS PERBANDINGAN *LOAD BALANCING WEB SERVER TUNGGAL* DENGAN *WEB SERVER CLUSTER MENGGUNAKAN LINUX VIRTUAL SERVER*”.
- [6] Microsoft. (2013). *Server Clustering*. Retrieved from <https://msdn.microsoft.com/enus/en%20us/library/ff649250.aspx>.
- [7] Mukmin, C. (2014). *IMPLEMENTASI CLUSTERING MENGGUNAKAN SHARE DISK DISTRIBUTED REPLICATED BLOCK DEVICE*. (S1), Universitas Bina Darma.