Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

HIGH AVAILABILITY DYNAMIC SHARDING DATABASE SERVER DENGAN METODE FAIL OVER DAN CLUSTERING

Afirda Desember Riawati[1], M Irfan[2], Khaeruddin[3,] Amrul Faruq[4]

[12,]Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Malang
[3,4,]Program Studi D3 Teknologi Elektronika, Fakultas Teknik, Universitas Muhammadiyah Malang

Jl. Raya Tlogomas No. 246 Malang - Jawa Timur 65144
[1] afirda@webmail.umm.ac.id, [2] irfan@umm.ac.id, [3] lamone@umm.ac.id, [4]faruq@umm.ac.id

Abstract

NoSQL is a type of database that is currently developing, which has a special design in solving problems related to scalability and reliability. The document format is a document-oriented database that is supported by the NoSQL database type. MongoDB is a document-oriented DBMS, MongoDB has a GridFS feature that can store data in binary form. Sharding can distribute data to multiple machines/servers, this function is also included in the MongoDB feature. Placing data on multiple machines makes it possible to store more data and handle larger loads without the need for a high-performance machine. High Availability, one of which is the ability of a system or group of systems (cluster) to keep the application or service running. The results obtained in this study are that a NoSQL-based database sharding system using MongoDB has been successfully implemented on a server that is set with High Availability, the main function of the server is for failover and clustering.

Keywords: *Sharding, Database, High Availability, Failover, Clustering*

Abstrak

NoSQL merupakan salah satu jenis database yang berkembang saat ini, yang mempunyai desain khusus dalam memecahkan permasalahan yang berhubungan dengan *scalability* dan *reliability*.Format dokumen adalah salah satu jenis *documentorienteddatabase* yang didukung oleh jenis database *NoSQL*.*MongoDB* merupakan salah satu jenis *DBMS* berbentuk *documentoriented*, *mongoDB* memiliki fitur *GridFS* yang dapat menyimpan data dalam bentuk biner.*Sharding* dapat mendistribusikan data ke dalam beberapa mesin/server, fungsi ini juga terdapat didalam fitur *mongoDB*.Dengan menempatkan data pada beberapa mesin memungkinkan untukmenyimpan lebih banyak data dan menangani beban lebih besar tanpa diperlukan adanya mesin denganperformansi tinggi. *High Availability* salah satunyamerupakan kemampuan satu sistem atau kelompok sistem (*cluster*) menjagaaplikasi (*application*) atau layanan (*service*) berjalan.Hasil yang didapat pada penelitian ini adalah sebuah system *sharding* database berbasis *noSQL* menggunakan *mongoDB* berhasil diimplementasikan pada server yang diset dengan *High Availability*, fungsi utama server adalah untuk *failover* dan *clustering*.

Kata kunci: *Sharding, Database,High Availability, Failover, Clustering*

1. PENDAHULUAN

Webmenjadi media berorientasibisnis dan antarmuka yang lebih disukai untuk sistem informasi terbaru[1]. Begitupesat menuntut semakin besar ukuran informasi yang berbentuk teks digital.Dokumen teks dengan jumlah yang sedikit, tentunya mudahbagi manusia untukmelakukan kategorisasi secara manual, namun pada dasarnya proses tersebutmenjadi suatu kondisi yang sulit dilakukan jika jumlah dokumennya mencapai ratusan hingga ribuan. Sehingga dibutuhkansebuah sistem yang dapat mengeloladan mengkategorikan dokumen dengan jumlah besar tersebut secara otomatis.Salah satu penggunaan yaitu server *Clustering* merupakan suatu opsi yang dapatdigunakan untuk mengatasi jika terjadi permasalahan tersebut, yaitu suatuteknologi

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

yang menggabungkan beberapa server yang bekerja bersama-sama yangseolah- olah merupakan satu sistem tunggal[2].

Pada umumnya, sistem kategorisasidokumen hanya sampai pada tahapkategorisasi saja. Pada sistem yang sudah ada dokumen hasil kategorisasi tersebutke dalam basisdata agar pengolahan data dapat dilakukan dengan mudah. Trenbasis data yang berkembang saat ini yaitu NoSQL yang didesain khusus untukmemecahkan permasalahan *scalability* dan *reliability*. Salah satu jenis dari NoSQLadalah document *oriented* database yang menyimpan data dalam format dokumen.Salah satu DBMS document *oriented* adalah MongoDB. MongoDB memiliki fitur *GridFS* yang dapat menyimpan data dalam bentukfile biner[2].

Selain itu, MongoDBjuga memiliki fitur sharding yang dapat mendistribusikan data ke dalam beberapamesin. Dengan menempatkan data pada beberapa mesin memungkinkan untuk menyimpan lebih banyak data dan menangani beban lebih besar tanpa diperlukan adanya mesin dengan performansi tinggi. *High* Availability salah satunya merupakan kemampuan satu sistem atau kelompok sistem (*cluster*) menjagaaplikasi (application) atau layanan (*service*) berjalan[3][12].

2. TINJAUAN PUSTAKA DAN TEORI

2.1. Tinjauan Pustaka

High Availability merupakan server berbeda berkerja bersama-sama untuk memastikan downtime pada resource dikurangi seminimal mungkin. Tujuan dari High Availability Cluster adalah untuk memastikan bahwa sumber daya mencapai Availability semaksimal mungkin [3], Jika server down atau jika sumber daya yang berhenti, HA cluster akan memonitor dan memastikan resource atau sumber daya dihidupkan ulang pada tempat lain dalam sistem cluster, sehingga dapat digunakan lagi setelah mengalami gangguan.

Konsep tersebut berkaitan dengan kemampuan sistem untuk mengatasi terjadinya gangguan, kerusakan hardware, crash/down, kesalahan jaringan bahkan kegagalan server yang di sebabkan software yang gagal melakukan tugas semestinya. Solusi yang ditawarkan berupa backup data atau failover data yang dilakukan secara real time. Saat server utama berhenti berjalan, maka server slave akan mengambil alih peran server utama dengan kualitas penanganan input dan output yang sama dengan server utama. Sistem akan selalu melakukan sinkronisasi data diantara keduanya atau mungkin lebih untuk mendapatkan redundancy data [4].

Failover merupakan sistem komunikasi dua atau lebih server ditempat yang berbeda yang dapat saling backup data dan mampu menggantikan pelayanan bila server lain down. Failover bertujuan untuk membantu menjaga akses client ke sumber daya di server, ketika server mengalami kegagalan fungsi software, atau kegagalan akses server. Failover cluster merupakan sekumpulan server yang saling bekerjasama untuk memberikan pelayanan meskipun berada ditempat yang berbeda, dan memiliki kualitas data atau sumberdaya yang sama antara server yang satu dengan server lainnya. Sistem Failover akan bekerja untuk menghubungi server-server yang menjadi anggota cluster-nya untuk mengambil alih tugas server utama saat terjadi kegagalan fungsi dalam waktu tertentu [5][11].

Pada tahun 1998 pertama kalinya dikembangkan sebuah konsep penyimpanan basis data yaitu NoSQL oleh Carlo Strozzi, yang kemudian pada tahun 2009 Eric Evans memperkenalkan kembali teknologi NoSQL. Kehadiran NoSQL bukan berarti untuk menggantikan model RDBMS yang sudah ada. Awal kemunculannya dilatarbelakangi oleh beberapa masalah yang muncul dari RDBMS. NoSQL dan RDBMS memiliki kelebihan dan tempat masing-masing sehingga diharapkan dapat saling melengkapi teknologi penyimpanan basis data [6]. Pada MongoDB dokumen disimpan dalam bentuk BSON (Binary JSON) [7]. Dengan struktur yang mirip dengan JSON membuatnyanya cukup mudah untuk dibaca.

Database NoSQL terbukti unggul dalam proses transaksi CRUD daripada SQL namun lemah untuk menghadapi transaksi join/agregasi , database NoSQL mampu memenuhi kebutuhan aplikasi ERP Retail yang bisa mensupport perbedaan skema dari setiap tenant, database NoSQL bersifat scalable[8].Tujuan daripada penelitian ini adalah implementasi dan membuktikan efektifitas teknologi virtualisasi Docker terintegrasi dengan openstack dalam hal penggunaan storage dan memory.

Ide penelitian ini didasarkan pada referenesi [3][4] tentang high availability dan fail over [5] yang diimplementasikan ke system database berbasis NoSQL [6][7][8] khususnya mongoDB. Penelitian ini membuktikan bahwa High

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

Availability Dynamic Sharding Database Server dengan metode Fail Over dan Clustering berhasil diimplementasikan.

2.2. *High available* (HA)

High available (HA) adalah sistem *cluster* yang terdiri dari dua buah server ataulebih mesin server yang biasa dikenal dengan node. Semua node yang terhubung dalam network saling berkomunikasi dan terhubung, pada setiap node pada clusterberkomunikasi dan menyampaikan informasi koneksi melalui *heart-beat*.

Cluster jugamemiliki penyimpanan untuk menyimpan data dan data tersebut terhubung padajaringan publik.Sebuah sistem *cluster* dapat menahan kegagalan perangkat keras dan perangkatlunak pada salah satu node dengan waktu *downtime* yang sangat kecil. *Cluster*dikonfigurasi untuk mendukung *high availability* dan *up time* pada server. HA cluster terdiri dari beberapa komponen hardware dan software yang rumit, dimana komponen *software*-nya adalah *operating system*, volume manager dan database[3].

2.2.1 Continuous Availability

Continuous availibility merupakan sistem yang dimana didalamnya memilikiservice yg dituntut agar sistem selalu pada kondisi available.Ini bertujuan untuk membuat sistem selalu siap dan tidak terjadi down. Untuk menjalankan sistem dibutuhkan 3 standar yang dapatdigunakan sebagai implementasi yaitu *Programmatic, Shared Availability, dan Multiple Copy[5].*

2.2.2 *Failover Availability*

Proses*failover*dan*failure*membutuhkan waktu yang sedikit saat terjadi down pada sistem, dimana didalam system menggunakan 2 buah serverdengan data identik pada masing-masing server. Sistem yang menggunakan konsep ini berjalan normal tanpa ada kendala, hanya saja pada master server yang melakukan suatu proses pada seluruh *user* pada suatu proses. Master server mengalami proses *failure* dan slave server menanggapi proses tersebut, slave server akan mengganti fungsi dari master server sehingga sistem server saling melakukan *back up* satu sama lain.

2.3. MongoDB

MongoDB merupakan salah satu produk open source yangbanyak digunakan dimana sistem ini menggunakan struktur data JSON untuk prosestersimpannya data. MongoDB juga sering oleh desainer sistem untuk penggunaanaplikasi dengan sistem yang terkoneksi cloud computing, grid computing atau big data[9].

MongoDB yang merupakan suatu sistem database dengan istilah open source berbasis dokumen yang dapat digunakan oleh semua *user* tanpa perlu *license,* MongoDBawalnya dibuat menggunakan bahasa C++ dan telahdikembangkan oleh 10gen sejakoktober 2007. Dokumen yang tersimpan didalamMongoDB dapat memiliki atribut yang berbeda dengan dokumen lain walaupun berada dalam satu koleksi[6].

MongoDB yang merupakan salah satu produk NoSQL juga dibagi menurut formatpenyimpanan data yaitu sebagai berikut :

a) Document database dimana setiap satu objek data yang ada dapat disimpankedalam dokumen-dokumen, kunci-kunci ataukeyvalue, dan *value/*nilai dapat berupa jenis data array.
b) Graph merupakan penyimpanan data dalam bentuk graph yang seringdigunakan dalam suatu interaksi seperti jejaring sosial.
c) Key-value atau kunci terenskrip pada database ini salah satu contohnya adalahapachecassandra.
d) Objek database merupakan suatu format dari database yang telah tersimpankedalam suatu objek[10].

3 METODOLOGI PENELITIAN

Dalam perancangan system inipembahasannya meliputi proses instalasi,konfigurasi dan integrasi perangkat lunak agar dapat berjalan pada jaringan yangdirancang. System sharding database yang dirancang adalah menggunakan empatunit server, dimana dua unit server difungsikan sebagai server *Shard* yang digunakan untuk menyimpan semua data.*Config Server* yang digunakan untuk menyimpan metadata klaster, dan server *Mongos/Query Router* yang digunakan untuk antarmukaaplikasi.

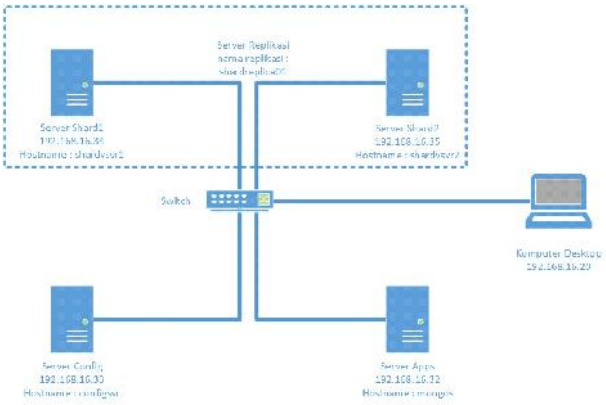
Aplikasi database yang digunakan dalampenelitian ini adalah MongoDB yangsistem basis datanya berorentasi dokumen dan dikembangkan oleh MongoDB Inc,aplikasi ini sifatnya open source dan gratis, aplikasi ini mempunyai performansikinerja yang tinggi, ketersediaan tinggi, dan penskalaan otomatis. Banyak
digunakan oleh instansi maupun perusahaan

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

yang mempunyai sistem basis datayang kompleks. Pada penelitian ini web server yang digunakan adalah *httpd*, webbased managemen basis data yang digunakan adalah *Mongo Express* dengan
diintegrasikan dengan PHP5.

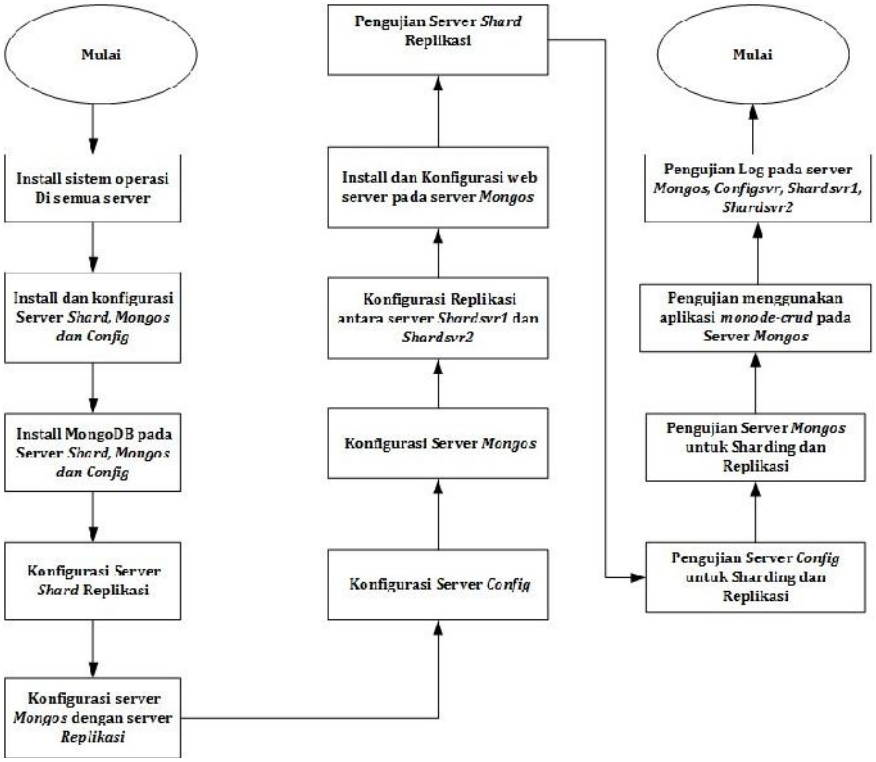
Tabel 1 IP Address Sistem

No	IP Server	Host Server	
1.	192.168.10.30	Mongos	Mongos
2.	192.168.10.31	Configsvr	Configsvr
3.	192.168.10.32	Shardsvr1	Shardsvr1
4.	192.168.10.33	Shardsvr2	Shardsvr2



Gambar 1. Topologi Jaringan High Availability Dynamic Sharding Database Server dengan

Pada penelitian ini menggunakan sistem operasi linux dan terinstall disemua server, linux yang digunakan adalah yang berbasis server. Adapun sistem operasi linux yang digunakan adalah distro turunan redhat yaitu CentOS 7 dan merupakan versi yang sesuai dengan kebutuhan penelitian.



Gambar 2. Rancangan pembangunan sistem High Availability Dynamic Sharding Database Server dengan metode Fail Over dan Clustering

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

Topologi jaringan yang digunakan dalam Penelitian ini adalah sepertiberikut,dimana ada dua buah server yang difungsikan sebagai serversharding dengansistem replikasi, satu server difungsikan sebagai server config dan satu serverdifungsikan sebagai server apps.

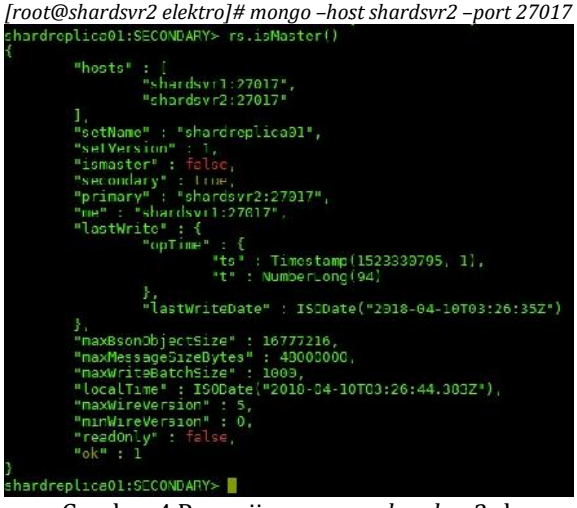
4 HASIL DAN PEMBAHASAN

Untuk mengetahui instalasi dan konfigurasi yang dilakukan sudah sesuai, maka pengujian keseluruhan sistem dilakukan pertahap. Pengujian tahap pertahap adalah memastikan system replikasi pada server berfungsi dengan semestinya, didalam system replikasi ada server yang berfungsi sebagai master dan slave.

[root@shardsvr1 elektro]# mongo –host shardsvr1 –port 27017

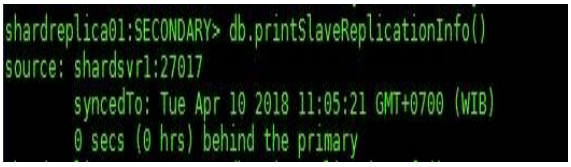
Gambar 3.Pengujian server *shardsvr1*master

[root@shardsvr2 elektro]# mongo –host shardsvr2 –port 27017



Gambar 4.Pengujian server *shardsvr2* slave

Pada pengujian antara server *shardsvr1* dan *shardsvr2* yang berfungsi sebagai server master adalah server *shardsvr2* dan untuk slavenya adalah server *shardsvr1.* Jika computer server direstart atau dimatikan fungsi *slave* dan *master* ini kadang-kadang berubah-ubah, kadang server *shardsvr1* akan berfungsi sebagai *master* dan *shardsvr2* sebagai *slave.* Tetapi fungsinya sebagai server database tetap berjalan dengan normal.



Gambar 5.Pengujian server *shardsvr2* slave perintah*db.printSlaveReplicationInfo()*

Informasi server slave, dimana pada pengujiandengan mengetikan perintah*db.printSlaveReplicationInfo()* server *slave* nya adalah *shardsvr1* dan terjadi syncronisasi dengan server master *shardsvr2* pada jam 11:05 WIB.

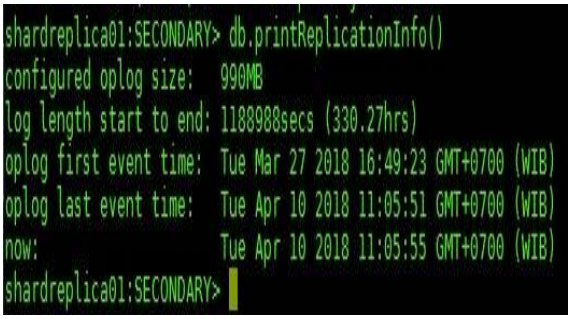
Gambar 6.Pengujian sinkronisasi server *shardsvr2* slave

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

Jumlah database yang disinkronisasikan antara *slave* dan *master* adalah 990MB dan terjadi sinkronisasi pada jam 11:05 WIB. Pengujian tahap kedua adalah memastikan system sharding pada server berfungsi dengan semestinya. Untuk melakukan pengujian sharding dilakukan di server *mongos* dan juga di server *shardsvr1 shardsvr2*.Kenapa dilakukan juga di server *mongos,* karena pada saat penambahan server sharding dilakukan di server *mongos.*Server *mongos* juga adalah server pengontrol sistem sharding dan replikasi database mongodb. Juga server *mongos* adalah server untuk menyimpan web aplikasi.Pada pengujian berikutdiperlihatkan *Sharding Status* pada server *mongo* diperlihatkan versi sharding nya adalah *version 6,* dan untuk *shards* nya adalah *shardreplica01* dan host nya ada *shardsvr1, shardsvr2* dengan port *27017,* dimana *active mongoses* nya dengan *version 3.4.13*

Gambar 7.Pengujian *Sharding Status* pada server *mongo*

Pada bagian database terdapat beberapa database yang telah dibuat, disini menggunakan database *monode-crud* dimana primary sharding nya adalah *shardreplica01.*



Gambar 8.Pengujian database *Sharding Status* pada server *mongo*

Pengujian server *configsvr* tetapi tetap diset replikasinya, untuk master dan slavenya hanya *configsvr* itu sendiri, seperti terlihat primary nya adalah 192.168.16.33 begitupun slavenya.
[root@configsvr elektro]# mongo –host configsvr –port 27019

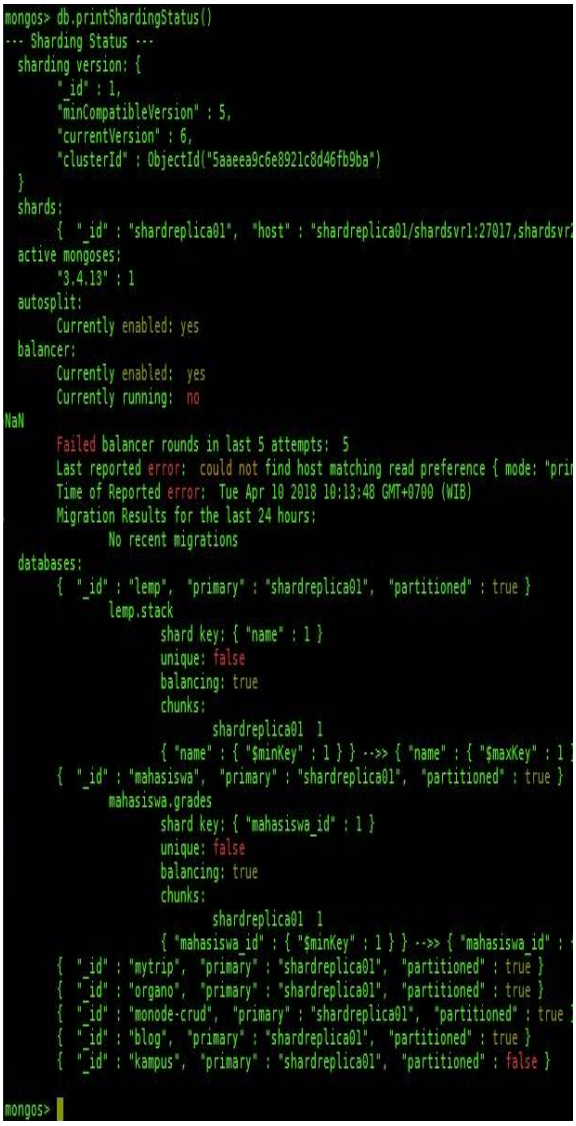
Gambar 9.Pengujian server *configsrv*

Pengujian pada server *mongos,* server ini membaca konfigurasi ke *configsvr* selain membaca konfigurasi yang berada pada server *mongos* sendiri juga membaca konfigurasi ke server *configsvr.* Untuk melihat database yang ada pada server ketikan command *show dbs,* akan muncul beberapa database yang sudah dibuat, pada pengujian ini menggunakan database *monode-crud.*

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

Gambar 10.Database pada server

Pengujian ini berfungsi untuk melihat status sharding pada server *mongos* terlihat bahwa server *mongos* membaca konfigurasi yang ada pada server *shardreplica01.*



Gambar 11.Database pada server

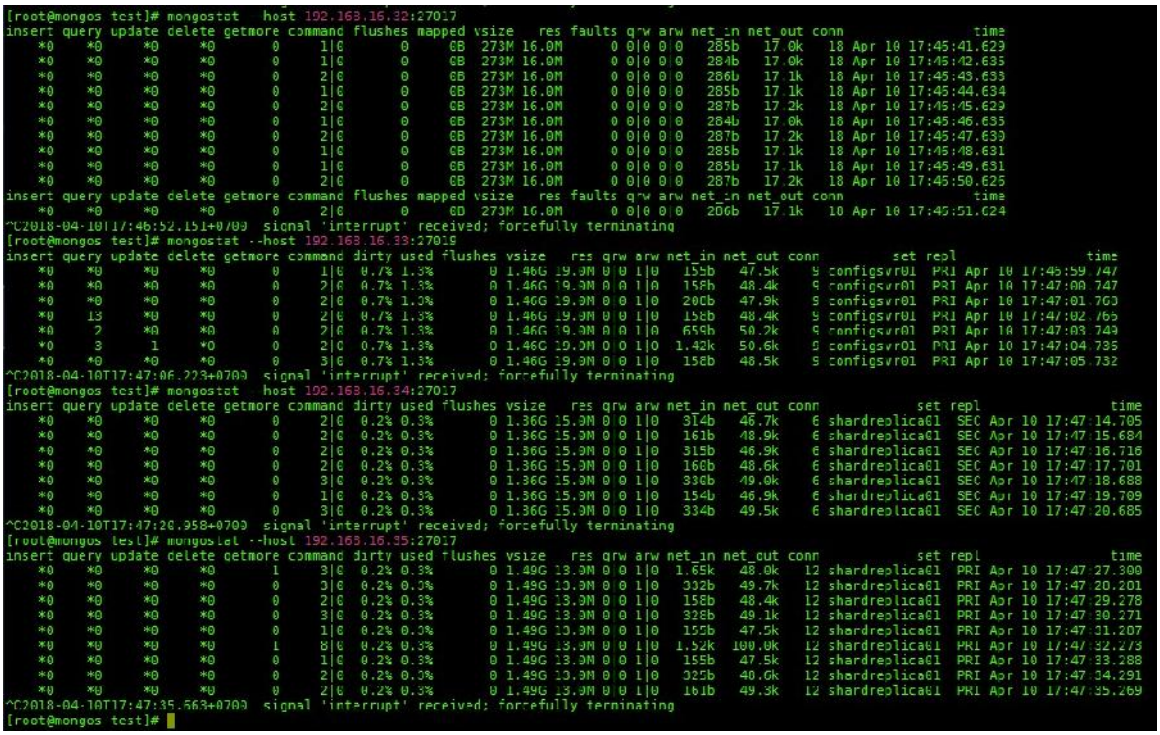
Gambar 12.Database pada server

Melihat stats dari database *monode-crud* menggunakan *db.stats()* terlihat bahwa database *monode-crud* disimpan di server replikasi *shardreplica01/shardsvr1:27017,shardsvr2:2701.* Pada pengujian tahap ketiga adalah untuk memastikan bahwa semua server yang terkoneksi dalam sistem sharding database bisa dimonitoring. Pada pengujian menggunakan mongostat terlihat terjadi koneksi antara server *mongos* dengan *localhost:27017* dan server *shardsvr1, shardsvr2* dan sistem replikasinya *shardreplica01.*

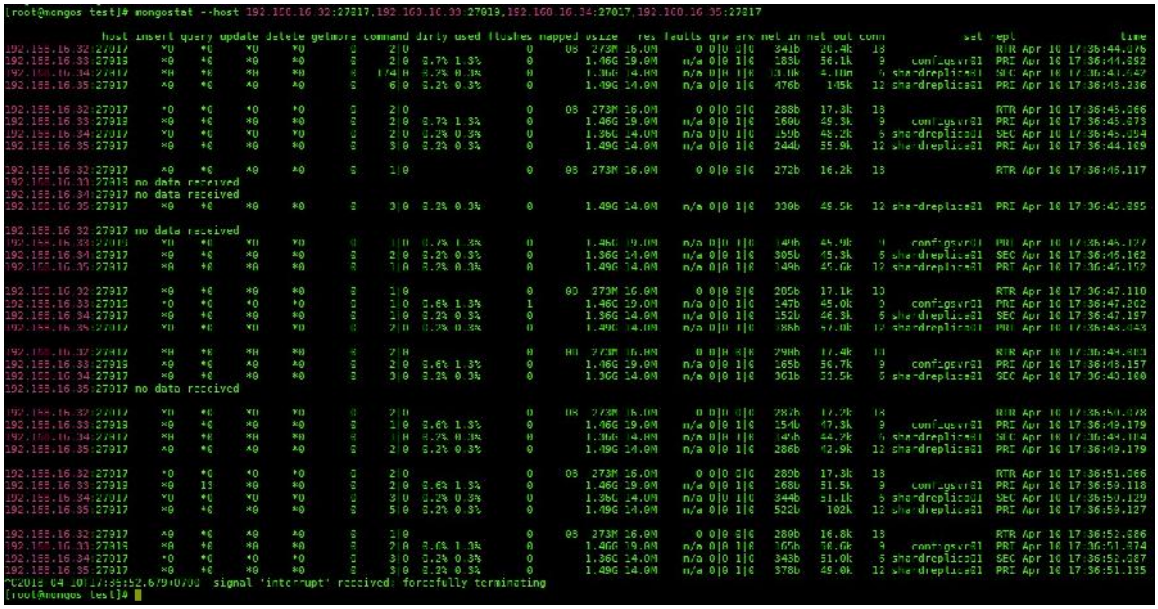
Pada pengujian *mongostat* dengan host yang lebih spesifik terlihat adanya koneksi antara server *mongos, configsvr, shardsvr1, shardsvr2* dengan ditandai adanya proses *insert, query, update, delete.* Sedangkan jika terlihat *no data received* ini menandakan tidak adanya proses *insert, query, update, delete* antara server.

Pada pengujian dengan menggunakan *mongostat* untuk tiap host bisa dilihat, untuk server *mongos* dengan ip *192.168.16.32:27017* terlihat tidak ada menggunakan replikasi, untuk server *configsvr* dengan ip *192.168.16.33:27019* terlihat menggunakan replikasi dengan set name *configsvr01,* untuk server *shardsvr1* dengan ip *192.168.16.34:27017* dan *shardsvr2* dengan ip *192.168.16.35:27017* menggunakan replikasi dengan set name *shardreplica01.*

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi



Gambar 13. proses *insert, query, update, delete* termonitoring di server mongos



Gambar 14. Status replikasi dan sharding termonitoring di server mongos

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

5 KESIMPULAN DAN SARAN

Pada penelitian ini, implementasi openstack terintegrasi docker telah berhasil dijalankan.Hasil yang didapat pada penelitian ini adalah sebuah system sharding database berbasis noSQL menggunakan mongoDB berhasil diimplementasikan pada server yang diset dengan *High* Availability, fungsi utama server adalah untuk fail over dan clustering.

6 UCAPAN TERIMA KASIH

Penulis dan tim peneliti mengucapkan terima kasih kepada Fakultas Teknik Universitas Muhammadiyah Malang (FT UMM), atas dukungan pendanaan melalui skema penelitian Pusat Kajian dan Rekayasa (Puskareka) FT UMM tahun 2020-2021. Juga kepada Laboratorium Jaringan Komputer Program Studi Teknik Elektro dan Laboratorium Jaringan Komputer Program Studi D3 Teknologi Elektronika UMM atas dukungan fasilitas sehingga terlaksana kegiatan penelitian ini.

DAFTAR PUSTAKA:

[1] Calzolari, Federico; Arezzini, Silvia; Ciampa, Alberto eds. High Availability Using Virtualization. IOP Sciense, 2010.

[2] Lwin dan Thein. High Availaibility Cluster System for Local Disaster Recovery with Markov Modeling Approach. International Journal of Computer Science Issues, 2009.

[3] Patil, N.V., et al. (2014). Cost Effective Failover Clustering. International Journal of Research in Engineering and Technology,2014.

[4] Michael Meadway, storing and retrieving objects on a computer network in a distributed database, 2014.

[5] Rasian, R dan Mursanto, P. Perbandingan Kinerja Pendekatan Virtualisasi. Jurnal Sistem Informasi. Magister Teknik Informatika Univeristas Indonesia,2009.

[6] Jones, M. Tim. High Availability with the Distributed Replicated Block Device. DeveloperWorks, 2014.

[7] Eelco Plugge, Peter Membrey dan Tim Hawkins, The Definitive Guide to MongoDB. Appress. 2010.

[8] Ameya Nayak, Anil Poriya, and Dikshay Poojary . Type of NOSQL Databases and its Comparison with Relational Databases Dept. of Computer Engineering Thakur College of Engineering and Technology University of Mumbai, March 2013.

[9] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira, and J. Bernardino, “Choosing the right NoSQL database for the job: a quality attribute evaluation,” J. Big Data, vol. 2, no. 1, p. 18, Aug. 2015.

[10] Divya Chauhan and K. L. Bansal ‘Using the Advantages of NOSQL: A Case Study on MongoDB’ Himachal Pradesh University Summerhill,Shimla,India.”

[11] Iqbal A.U, Nurhadi, Lailis S., Khaeruddin; ‘Implementasi System Aplikasi Docker Terinttegrasi Openstack’ Vol. 4 No. 1 (2021): JIRE April 2021.

[12] Yuliadi, Nurul M. S., Herfandi; ‘Rekayasa Aplikasi Center Rumah Kost Berbasis Web Di Kabupaten Sumbawa’ Vol. 4 No. 2 (2021): MISI Juni 2021.

[13] Imtihan, K., & Fahmi, H. (2020). Analisis Dan Perancangan Sistem Informasi Daerah Rawan Kecelakaan Dengan Menggunakan Geographic Information Systems (GIS). Jurnal Manajemen Informatika dan Sistem Informasi, 3(1), 16-23.

[14] Imtihan, K., & Basri, M. H. (2019). Sistem Informasi Pembuatan Manifest Muatan Kapal Berbasis Dekstop Dan Android. Jurnal Manajemen Informatika dan Sistem Informasi, 2(2), 69-76.

[15] Ashari, M., Zaen, M. T. A., Putri, J. A., Imtihan, K., & Bagye, W. (2022). Prototype Sterilisasi Virus Barang Belanjaan Online Berbasis Arduino. Jurnal Media Informatika Budidarma, 6(1), 120-127.

[16] Imtihan, K., Bagye, W., Asri, Z. M. T., Fadli, S., & Ashari, M. (2021, February). Image

Jurnal Manajemen Informatika & Sistem Informasi (MISI)
Volume 5, Nomor 1, Januari 2022.
ISSN 2614-1701 (media cetak)
ISSN 2614-3739 (media online)
DOI : 10.36595/misi.v5i1
http://e-journal.stmiklombok.ac.id/index.php/misi

capture device based on Internet of Thing (IoT) technology. In IOP Conference Series: Materials Science and Engineering (Vol. 1088, No. 1, p. 012065). IOP Publishing

[17] Imtihan, K., Mutawalli, L., & Bagye, W. (2020). Pemilihan Model Scrum Dalam Pengembangan Sistem Monitoring Dengan Menggunakan Metode Agile Untuk Evaluasi Clinical Pathway. Bianglala Informatika, 8(1), 63-69.