

**ANALISIS DAN PERBANDINGAN LOAD BALANCER DATABASE HAProxy
DAN MAXSCALE DALAM PERFORMA KECEPATAN WEBSITE E-LEARNING
MOODLE**

SKRIPSI



Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Sarjana Pendidikan

Disusun oleh:

Anjar Fiandrianto

NIM. 08520244014

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2015**

HALAMAN PERNYATAAN KEASLIAN

LEMBAR PERSETUJUAN

Saya yang bertandatangan di bawah ini

Nama

Tugas Akhir Skripsi dengan Judul

PERBANDINGAN LOAD BALANCER DATABASE HAProxy DAN MAXSCALE
DALAM PERFORMA KECEPATAN WEBSITE E-LEARNING MOODLE

Mengetahui dan Mewacana dalam Performa Kecepatan

Disusun oleh:

Anjar Fandrianto

NIM. 08520244014

menyatakan bahwa skripsi ini benar-benar hasil penelitian dan karya
pribadi saya tidak pernah diberikan atau diberikan pengetahuan atau
penulisan karya lainnya.

Telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan
Ujian Tugas Akhir Skripsi bagi yang bersangkutan.

Yogyakarta, 10 September 2015

Mengetahui,
Ketua Program Studi
Pendidikan Teknik Informatika

Muhammad Munir, M.Pd
NIP. 19630512 198901 1 001

Disetujui,
Dosen Pembimbing

Handaru Jati, Ph.d
NIP. 19740511 199903 1 002

NIM. 08520244014

HALAMAN PERNYATAAN KEASLIAN

Saya yang bertandatangan di bawah ini:

Nama : Anjar Fiandrianto

NIM : 08520244014

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Analisis dan Perbandingan *Load Balancer* Database

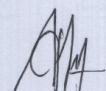
 Haproxy dan Maxscale dalam Performa Kecepatan

 Website E-Learning Moodle

menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang dapat ditulis atau diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Yogyakarta, September 2015

Yang menyatakan,



Anjar Fiandrianto

NIM. 08520244014

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

ANALISIS DAN PERBANDINGAN LOAD BALANCER DATABASE HAProxy DAN MAXSCALE DALAM PERFORMA KECEPATAN WEBSITE E-LEARNING MOODLE

Disusun oleh:

Anjar Fiandrianto

NIM. 08520244014

Telah dipertahankan di depan tim penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta

Pada tanggal September 2015

TIM PENGUJI

Nama / Jabatan

Tanda Tangan

Tanggal

Handaru Jati Ph.d /

Ketua Penguji (Pembimbing)  19-10-2015

Totok Sukardiyono, MT /

Sekretaris

19-10-2015

Dr. Eko Marpanaji /

Penguji Utama

19-10-2015



HALAMAN MOTTO

"One day you'll leave this world behind,
so live a life you will remember"

Avicii

HALAMAN PERSEMPAHAN

Alhamdulillah, segala puji bagi Allah SWT atas segala karunia-Nya sehingga penulis mampu menyelesaikan penulisan skripsi ini. Skripsi ini penulis persembahkan untuk Ibu saya yang sabar menunggu saya menyelesaikan skripsi ini.

ANALISIS DAN PERBANDINGAN LOAD BALANCER DATABASE HAProxy DAN MAXSCALE DALAM PERFORMA KECEPATAN WEBSITE E-LEARNING MOODLE

Oleh:
Anjar Fiandrianto
NIM. 08520244014

ABSTRAK

Tujuan penelitian ini dirancang untuk: (1) Mengetahui cara membuat database cluster untuk meningkatkan skalabilitas Moodle, (2)mengetahui cara menghilangkan deadlock yang terjadi di database cluster yang menggunakan HAProxy, (3) mengetahui perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale.

Penelitian ini merupakan penelitian deskriptif komparatif untuk menganalisis dan meyimpulkan apakah ada perbedaan performa saat database cluster yang digunakan di moodle saat menggunakan HAProxy atau MaxScale. Tempat penelitian menggunakan layanan EC2 dari Amazon Web Service. Intrumen penelitian yang digunakan adalah perangkat lunak Htpprof dengan berdasar pada standar internasional ITU-T G.1030 11/2005. Variabel yang diteliti di penelitian ini adalah Troughput, Response Time, Reply Client dan Error Client.

Hasil penelitian ini diketahui bahwa: (1) Moodle bisa dijadikan sistem cluster dengan menjadikan web server tunggal menjadi cluster dan database server tunggal menjadi database cluster menggunakan Galera Cluster dan Percona Cluster, (2) load balancer MaxScale mempunyai fitur read-write split yang menjadikan database server tidak mengalami deadlock sehingga database terbebas dari masalah redudansi data, (3)tidak ditemukan perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale.

Kata kunci : cluster, ITU-T G.1030 11/2005, haproxy, maxscale, aws

ANALYSIS AND COMPARATIVE HAProxy AND MAXSCALE DATABASE LOAD BALANCER SPEED PERFORMANCE IN MOODLE E-LEARNING WEBSITE

By:
Anjar Fiandrianto
NIM. 08520244014

ABSTRACT

The purpose of this research was designed to: (1) Determine how to create the cluster database to improve Moodle scalability, (2) determine how to eliminate deadlocks that occur in the database cluster using HAProxy, (3) determine differences performance Moodle website if using a load balancer HAProxy or MaxScale,

This research is a comparative descriptive research to analyze and concludes whether there are differences in performance when the database cluster used in Moodle when using HAProxy or MaxScale. This research using the services of Amazon Web Services EC2. Instruments used in this study is software Httperf based on an international standard ITU-T G.1030 11/2005. Variables examined in this study is throughput, response time, reply client and client error.

Results of this research: (1) Moodle can be used as the cluster system by making a single web server into a cluster and database single server into a database cluster using Galera Cluster and Percona Cluster, (2) load balancer MaxScale features the read-write split that makes the database server is not experiencing a deadlock so that the database is free from the problem of data redundancy, (3) no differences were found when using the Moodle website performance load balancer HAProxy or MaxScale.

Index Terms : cluster, ITU-T G.1030 11/2005, haproxy, maxscale, aws

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka untuk memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan judul "Analisis dan Perbandingan *Load Balancer* Database Haproxy dan Maxscale dalam Performa Kecepatan Website E-Learning Moodle" dapat disusun sesuai dengan harapan. Tugas akhir skripsi ini dapat diselesaikan tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkennaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Handaru Jati, Ph.d; selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan tugas akhir skripsi
2. Tim penguji yaitu Ketua Penguji, Sekertaris, dan Penguji yang memberikan koreksi perbaikan secara komprehensif terhadap TAS ini.
3. Muhammad Munir, M.Pd; selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesaiya Tugas Akhir Skripsi ini.
4. Dr. Moch. Bruri Triyono selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi.

5. Achmad Fatchi M.Pd, selaku Dosen Penasihat Akademik yang selalu memberikan arahan dan bimbingan dalam menempuh studi ini.
6. Kedua orang tua, Bapak dan Ibu serta keluarga yang senantiasa memberikan dukungan, motivasi dan doa.
7. Semua pihak, secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini.

Akhirnya, semoga segala bantuan yang telah diberikan semua pihak di atas menjadi amalan yang bermanfaat dan mendapatkan balasan dari Allah SWT dan tugas akhir skripsi ini menjadi informasi bermanfaat bagi pembaca atau pihak lain yang membutuhkannya.

Yogyakarta, September 2015

Penulis,

Anjar Fiandrianto

NIM. 08520244014

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
HALAMAN PERNYATAAN KEASLIAN	iii
HALAMAN PENGESAHAN	Error! Bookmark not defined.
HALAMAN MOTTO	iv
HALAMAN PERSEMBAHAN	vi
ABSTRAK.....	vii
ABSTRACT	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
A. LATAR BELAKANG.....	1
B. IDENTIFIKASI MASALAH	5
C. PERUMUSAN MASALAH	6
D. TUJUAN PENELITIAN	6
E. MANFAAT PENELITIAN.....	7
BAB II KAJIAN PUSTAKA	8
A. KAJIAN TEORI.....	8

1. Linux Cluster.....	8
2. E-Learning.....	15
3. Load Balancer	16
4. Galera Cluster	18
5. Cloud Computing.....	20
6. Tahap Pengembangan Arsitekur Infrastruktur.....	26
7. Htpperf	31
8. Notasi Big-O	31
9. Variabel Performa Server Berdasar ITU-T G.1030 11/2005.....	32
B. KAJIAN PENELITIAN YANG RELEVAN	34
C. KERANGKA BERPIKIR.....	34
BAB III METODE PENELITIAN.....	36
A. DESAIN PENELITIAN.....	36
B. METODE PENELITIAN	36
C. PROSEDUR PENELITIAN.....	37
1. Mencari ide atau gagasan penelitian.....	37
2. Melakukan study literature	37
3. Menentukan rumusan masalah	37
4. Menentukan tujuan penelitian.....	37
5. Melakukan pengambilan data.....	38
6. Menganalisis data.....	38
7. Merumuskan kesimpulan	38

D. LOKASI PENELITIAN	38
E. POPULASI DAN SAMPEL PENELITIAN.....	39
1. Populasi Penelitian.....	39
2. Sampel penelitian	39
F. TEKNIK PENGUMPULAN DATA	41
G. INSTRUMEN PENELITIAN	41
H. TEKNIK ANALISIS DATA.....	42
BAB IV HASIL DAN PEMBAHASAN	43
A. TAHAP ANALISIS DAN DOKUMENTASI ARSITEKTUR LAMA	43
1. Analisis Kebutuhan Aplikasi.....	43
2. Analisis Kebutuhan Hardware	43
3. Analisis Kebutuhan Media Penyimpanan.....	43
4. Analisis Kebutuhan Jaringan	44
5. Analisis Kebutuhan Operasional	44
B. MENDESAIN ARSITEKTUR BARU	44
1. Analisis Data Arsitektur Lama.....	44
2. Analisis Kebutuhan Arsitektur Baru	45
3. Membuat Desain Arsitektur Dasar	46
4. Membuat Prototype Arsitektur	47
5. Revisi Prototype Arsitektur Baru	47
C. MEMBANGUN INFRASTRUKTUR MENGGUNAKAN ARSITEKTUR BARU	48
1. Menggunakan layanan EC2.....	49

2. Sistem Operasi	50
3. Spesifikasi Virtual Machine.....	50
4. Spesifikasi jaringan	51
5. Spesifikasi Storage	51
6. Konfigurasi keamanan Virtual Machine.....	52
7. Konfigurasi IP Publik.....	53
8. Konfigurasi Global Server	54
9. Database Server (Percona XtraDB Cluster)	54
10. Web Server (Apache Web Server dan PHP).....	55
11. Load Balancer (HaProxy dan MaxScale)	55
D. HASIL PENGUJIAN LOAD BALANCER.....	55
1. Hasil Pengujian Throughput.....	57
2. Hasil Pengujian Response Time.....	58
3. Hasil Pengujian Reply Client.....	59
4. Hasil Pengujian Error Client	60
E. ANALISIS DATA HASIL PENGUJIAN	61
1. Analisis Throughput.....	62
2. Analisis Response Time	63
3. Analisis Reply Client.....	64
4. Analisis Error Client.....	64
F. PEMBAHASAN HASIL PENELITIAN	65
1. Throughput.....	65

2. Response time	66
3. Reply Client	66
4. Error Client	66
BAB V SIMPULAN DAN SARAN	68
A. SIMPULAN	68
B. SARAN.....	68
DAFTAR PUSTAKA	70
LAMPIRAN	72

DAFTAR GAMBAR

Gambar 1. Cluster Sebagai "a single unified computing resource "	10
Gambar 2. Sistem Cluster Sederhana	10
Gambar 3. Jaringan Privat Cluster	13
Gambar 4. Sistem <i>Shared Disk</i> Cluster	14
Gambar 5. Arsitektur Load Balancer	17
Gambar 6. Cloud Computing	21
Gambar 7. Jenis-jenis Cloud Computing	22
Gambar 8. Jenis Cloud Berdasar Pengguna	23
Gambar 9. Karakteristik Cloud Computing	25
Gambar 10. Contoh Analisis Kebutuhan Aplikasi	26
Gambar 11. Arsitektur Lama Moodle	44
Gambar 12. Arsitektur Baru Moodle	46
Gambar 13. Arsitektur Baru Jaringan Moodle	47
Gambar 14. Arsitektur Moodle dalam Penelitian	48
Gambar 15. Halaman Layanan AWS	49
Gambar 16. Pilihan Sistem Operasi AWS	50
Gambar 17. Jenis-Jenis Virtual Machine di AWS	51
Gambar 18. Konfigurasi Jaringan	51
Gambar 19. Spesifikasi Storage VM AWS	52
Gambar 20. Konfigurasi Firewall VM AWS	53
Gambar 21. Detail Alamat IP VM AWS	54

Gambar 22. Proses Pengujian Load Balancer	57
Gambar 23. Hasil Pengujian throughput.....	58
Gambar 24. Hasil Pengujian Response Time.....	59
Gambar 25. Hasil Pengujian Reply Client.....	60
Gambar 26. Hasil Pengujian Error Client	61
Gambar 27. Hasil Analisis Throughput	62
Gambar 28. Hasil Analisis Response Time	63
Gambar 29. Hasil Analisis Reply Client	64
Gambar 30. Hasil Analisis Error Client.....	65

DAFTAR TABEL

Tabel 1. Hasil Pengujian Throughput.....	57
Tabel 2. Hasil Pengujian Response time.....	58
Tabel 3. Hasil Pengujian Reply Client.....	59
Tabel 4. Hasil Pengujian Error Client.....	60

DAFTAR LAMPIRAN

Lampiran 1. File Konfigurasi Server Database Server 1	73
Lampiran 2. File Konfigurasi Server Database Server 2	74
Lampiran 3. File Konfigurasi Server Database Server 3	75
Lampiran 4. File Konfigurasi Server HAProxy	76
Lampiran 5. File Konfigurasi Server MaxScale	78
Lampiran 6. File Konfigurasi Moodle	80
Lampiran 7. Halaman Akun Amazon Web Services	81

BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi Informasi telah berkembang sangat pesat, dimulai dengan sistem komputer *stand-alone* sampai saat ini semuanya telah dimungkinkan terhubung ke internet. Hampir seluruh sektor yang ada di sekitar kita telah memanfaatkan teknologi informasi untuk membantu pekerjaan sehari-hari ataupun sekedar untuk hiburan. Semakin berkembangnya pengguna internet, otomatis semakin besar juga data yang ditransmisikan dan data yang disimpan dalam jaringan autonom. Sekarang ini, paradigma telah berubah, pada sektor industri misalnya, pada jaman dahulu mesin fisik memegang peranan yang paling penting dalam industri, tapi saat ini data digital-lah yang menjadi hal yang paling penting di suatu industri. Saat ini hampir semua hal telah dibuat sistem online untuk menambah produktifitas dan kemudahan pendistribusian data. Hal itu menunjukkan bahwa saat ini begitu besar pengaruh keberadaan sebuah data dalam segala aspek.

Dunia pendidikan sekarang ini sangat cepat dalam mengadopsi teknologi informasi. Penggunaan teknologi informasi menjadikan proses belajar mengajar menjadi lebih hidup dan interaktif. Pada awal adopsi teknologi informasi, sebagian pengajar menggunakan teknologi informasi untuk menyampaikan presentasi materi menggunakan Power Point, Adobe Flash, maupun aplikasi khusus lain yang mempunyai fungsi yang sama. Saat ini seiring dengan tumbuhnya internet, proses belajar mengajar menjadi banyak memanfaatkan

internet, diantara menggunakan e-learning. Salah satu e-learning yang paling populer saat ini adalah Moodle, Moodle saat ini menjadi salah satu LMS (*Learning Management System*) yang paling populer menurut suurvei yang dilakukan oleh Capterra. Dalam infografis yang dirilis capterra, terlihat bahwa Moodle mempunyai sekitar 74 juta pengguna di seluruh dunia.

Proses pembelajaran menggunakan Moodle berarti data ada sebagian data pembelajaran ditampung di Moodle, Moodle akan menjadi sebuah sistem yang penting sehingga harus dibangun dengan arsitektur sistem yang baik. Cade dan Roberts menjelaskan bahwa sistem arsitektur yang baik harus memiliki 7 aspek yaitu, *scalability, reliability, availability, extensibility, maintainability, manageability* dan *security*. Diantara 7 aspek tersebut, *scalability* sangat penting diperhatikan karena sebuah sistem yang baik harus bisa tetap bekerja meski jumlah pengguna terus bertambah. Meningkatkan skalabilitas bisa diperoleh dengan peningkatan kapasitas perangkat keras ataupun optimalisasi perangkat lunak. Aspek skalabilitas juga diperlukan untuk LMS Moodle. Moodle harus bisa dikonfigurasi secara sistem agar tingkat skalabilitasnya tinggi. Ketika pengguna sebuah *website learning* bertambah banyak, Moodle tetap bisa beroperasi normal. Berdasarkan observasi masalah di SMP 1 Wates sebagai pengguna Moodle, website sering mengalami *overload* saat penggunanya melebihi kapasitas. Server menggunakan VPS dengan RAM 512MB, jika ingin meningkatkan kapasitas harus melakukan upgrade VPS.

Peningkatkan skalabilitas Moodle lebih mudah dengan penambahan kapasitas hardware yang digunakan untuk server Moodle daripada optimalisasi perangkat lunak. Penambahan kapasitas hardware dibagi menjadi 2 metode yaitu *scale-out*

(mendatar) dan *scale-up* (menurun). *Scale-out* berarti menambah atau mengurangi jumlah server disesuaikan dengan jumlah pengguna yang mengakses Moodle, sedangkan *scale-up* berarti melakukan upgrade komponen dalam satu server seperti CPU dan RAM. Moodle standar saat ini hanya bisa ditingkatkan skalanilitasnya menggunakan metode *scale-up* yaitu dengan melakukan *upgrade* server, CPU, RAM, atau media penyimpanan.

Keberhasilan peningkatkan skalabilitas Moodle melalui peningkatkan kapasitas *hardware* dengan metode *scale-out* ditentukan oleh jumlah *web server* dan *database server*. Semakin banyak *web server* dan *database server*, semakin banyak pengguna yang bisa mengakses Moodle secara bersamaan, sistem tersebut disebut sebagai sistem cluster. Namun di Moodle standar sampai dengan versi terbaru, tidak tersedia fitur untuk membuat sistem yang biasa menjadi sistem cluster. Sistem cluster mampu melakukan pembagian beban komputasi ke banyak server menggunakan sebuah *load balancer*. *Load balancer* untuk Moodle terbagi menjadi dua yaitu *web load balancer* dan *database load balancer*. *Web load balancer* dapat menggunakan nginx, apache, ataupun haproxy sedangkan untuk *database load balancer* dapat menggunakan HAProxy atau MaxScale.

HAProxy dan MaxScale keduanya merupakan *database load balancer*, namun memiliki karakteristik yang berbeda. HAProxy langsung mentranslasikan data tanpa harus melihat isi data, sedangkan MaxScale mentranslasikan data namun dengan fitur yang lebih detail sehingga harus melihat data yang lewat sebelum ditranslasikan ke server tujuan. Fitur MaxScale yang bisa melihat data (query) sebelum disampaikan ke server tujuan ini menjadi sisi positif karena bisa

membaca *query* yang *read* dan *write*. MaxScale memiliki fitur *read-write split*, yaitu memisahkan server yang digunakan untuk membaca dan menulis ke database server. Berdasarkan observasi masalah di PT sebangsa Bersama dan PT Pojok Celebes (Telkom Grup), sering muncul deadlock sehingga membuat munculnya redundansi *unique* data yang terjadi jika menggunakan HAProxy dengan database Percona XtraDB Cluster. Menggunakan fitur *read-write split* yang ada di MaxScale seharusnya bisa memecahkan masalah ini.

HAProxy pertama kali dirilis pada tahun 2001, sedangkan MaxScale pertama kali rilis pada akhir tahun 2014. MaxScale membawa beberapa fitur yang tidak ada di HAProxy diantaranya sharding dan read-write split. Fitur-fitur tambahan tersebut menggunakan algoritma yang lebih kompleks. Berdasarkan notasi Big-O, setiap penambahan algoritma akan menambah waktu yang dibutuhkan untuk pemrosesan data. Berdasar teori tersebut seharusnya kecepatan pemrosesan MaxScale lebih rendah karena menggunakan algoritma yang lebih kompleks dibandingkan menggunakan HAProxy yang algoritmanya lebih sederhana. Setelah dilakukan observasi awal oleh peneliti menggunakan website drupal, ternyata memang terjadi penurunan kecepatan di website drupal saat load balancer diganti dari HAProxy menjadi MaxScale. Saat ini belum ada ujicoba komparasi yang menunjukkan perbedaan performa penggunaan HAProxy dan MaxScale yang diimplementasi untuk Moodle.

Pemilihan *load balancer* yang tepat untuk Moodle sangat berpengaruh terhadap kecepatan server menampilkan sebuah halaman di browser. Untuk memilih load balancer mana yang tepat maka harus dilakukan uji coba terhadap kedua load balancer tersebut.

Permasalahan-permasalahan diatas menginspirasi penulis untuk melakukan penelitian dengan melakukan observasi untuk mengetahui berapa besar perbedaan performa kecepatan website e-learning Moodle saat menggunakan HAProxy dan MaxScala.

B. Identifikasi Masalah

Dari fakta-fakta pada latar belakang, dapat diidentifikasi beberapa masalah, diantaranya:

1. Peningkatan kapasitas Moodle standar tidak mendukung metode *scale-out* namun hanya bisa menggunakan metode *scale-up* yaitu menambahkan kapasitas CPU, RAM dan media penimpanan.
2. Moodle secara standar belum mempunyai sistem database cluster. Moodle berjalan di satu web server dan satu databaser server.
3. Load balancer HAProxy dengan database cluster sering mengalami masalah deadlock sehingga terjadi redundansi data yang seharusnya bertipe *unique*.
4. MaxScale mempunyai algoritma yang lebih kompleks dalam metode load balancing sehingga berdasar notasi Big-O ada asumsi performa akan menjadi lebih lambat.
5. Tidak diketahui apakah ada perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale.

B. Batasan Masalah

Berdasarkan masalah yang telah diidentifikasi, maka penelitian ini difokuskan pada penyelesaian masalah-masalah berikut:

1. Moodle secara standar belum mempunyai sistem database cluster. Moodle berjalan di satu web server dan satu databaser server.
2. Load balancer HAProxy dan database cluster sering mengalami masalah deadlock sehingga terjadi redudansi data yang seharusnya bertipe unique.
3. Tidak diketahui apakah ada perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale.

C. Perumusan Masalah

1. Bagaimana cara mengatasi masalah rendahnya skalabilitas pada database Moodle?
2. Bagaimana cara menghilangkan deadlock yang terjadi di database cluster yang menggunakan HAProxy?
3. Apakah ada perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale?

D. Tujuan Penelitian

Adapun tujuan dari penelitian ini, antara lain:

1. Mengetahui cara membuat database cluster untuk meningkatkan skalabilitas Moodle.
2. Mengetahui cara menghilangkan deadlock yang terjadi di database cluster yang menggunakan HAProxy.
3. Mengetahui apakah ada perbedaan performa website Moodle jika menggunakan load balancer HAProxy atau MaxScale.

E. Manfaat Penelitian

Manfaat dari penelitian ini, antara lain:

1. Manfaat Praktis

- a. Bagi instansi/ lembaga, penelitian ini dapat menjadi referensi dalam mengembangkan atau memilih *database load balancer sever*.
- b. Bagi peneliti, penelitian ini dapat meningkatkan pengetahuan penulis dalam bidang *database load balancer sever*.

2. Manfaat Teoritis

Hasil penelitian ini dapat menjadi referensi bagi penelitian selanjutnya yang topiknya berkaitan.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Linux Cluster

a) Pengertian dan Cara Kerja Linux Cluster

Linux cluster merupakan sebuah sistem cluster yang berjalan diatas sistem operasi Linux. Pfister (1997) memberikan definisi tentang cluster yaitu sebuah tipe sistem paralel dan sistem terdistribusi yang terdiri dari kumpulan komputer yang saling berhubungan dan berjalan layaknya sebuah sistem yang besar, walaupun didalamnya terdapat bagian bagian namun bagian tersebut tidak terlihat oleh pengguna, bahkan programmer. Penjelasan senada juga dijelaskan oleh Jayaswal (2006) dalam bukanya yang berjudul "Administering Data Centers: Servers, Storage, and Voice over IP", cluster didefinisikan sebagai kumpulan sistem yang saling terhubung namun bekerja sebagai satu sistem tunggal, bersifat *scalable*, dan *High Available*. Konsep cluster membuat sebuah sistem berjalan dengan *downtime* yang minimal. Keuntungan menggunakan sistem cluster adalah meningkatkan performa, pemenuhan *SLA* dan mempermudah pengelolaan sistem.

Syarat utama sebuah cluster adalah seluruh server harus berjalan sebagai sebuah sistem tunggal. Kopper (2005) menyebutkan agar sistem kita sukses berjalan sebagai sebuah sistem tunggal, ada empat karakteristik yang harus dipenuhi antara lain:

- 1) Pengguna tidak tahu bahwa mereka sedang menggunakan sebuah sistem cluster.

Karakteristik ini bisa dilihat di sistem-sistem yang besar seperti google. Di Google kita bisa yakin Google menggunakan cluster namun secara awam tidak bisa dibuktikan.

- 2) Setiap server yang ada di dalam cluster tidak tahu bahwa server tersebut bagian dari sebuah cluster.

Sistem operasi tidak perlu dikonfigurasi ulang untuk menjadi bagian dari sebuah cluster. Sebuah server saat mengalami gangguan juga tidak membuat semua anggota cluster menjadi bermasalah. Saat sebuah server mati, sistem harus tetap berjalan normal dengan downtime yang minimal.

- 3) Aplikasi yang berjalan di atas sebuah cluster tidak tahu jika mereka berjalan di atas sebuah cluster

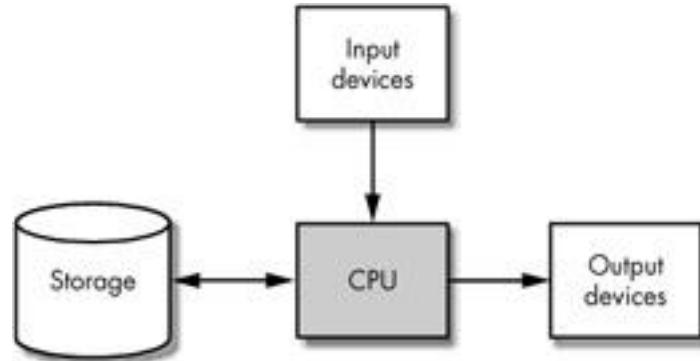
Jika kita membangun website dengan sistem cluster, seharusnya website tersebut tidak mengalami modifikasi yang terlalu banyak agar bisa berjalan diatas sistem cluster.

- 4) Server pendukung yang tidak secara langsung anggota cluster namun satu jaringan dengan cluster tidak tahu bahwa mereka terkoneksi dengan sistem cluster

Mail server dan DNS merupakan sebuah sistem yang di luar sistem cluster namun berfungsi untuk mendukung kinerja sistem cluster. Server pendukung seperti ini seharunya tidak tahu bahwa mereka sedang melayani sebuah sistem cluster. Walaupun server pendukung tidak terisolasi secara jaringan namun server-server tersebut terisolasi secara fungsi.

b) Arsitektur Cluster

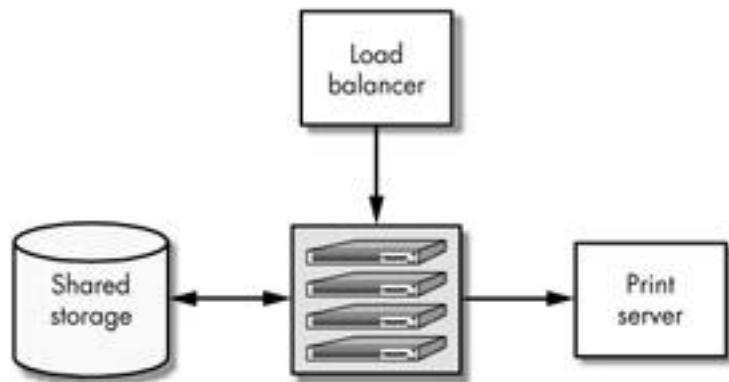
Sebuah sistem cluster dijelaskan oleh Pfister(1997) harus berfungsi seperti "*a single unified computing resource*". Gambar dibawah ini menjelaskan konsep tersebut:



Gambar 1. Cluster Sebagai "a single unified computing resource "

Sumber : Pfister(1997)

Kopper (2005) memberi contoh, jika konsep tersebut diimplementasikan kedalam sistem cluster, maka secara sederhana gambarnya akan sebagai berikut:



Gambar 2. Sistem Cluster Sederhana

Sumber : Kopper (2005)

Load balancer berfungsi sebagai input devices yang menerima request dari pengguna dan menyebarkan ke semua node anggota cluster. Anggota cluster jumlahnya menyesuaikan dengan jumlah request yang masuk setiap detik dan berapa beban yang timbul di node cluster untuk setiap requestnya. Menurut Jayaswal (2006) terdapat empat jenis cluster yaitu:

1) Performance Cluster

Performance cluster bertujuan untuk meningkatkan kinerja, skalabilitas dan *availability*. Server ini juga sering disebut sebagai high-performance computing clusters (HPPC) atau paralel cluster.

2) Fail-over Cluster

Fail-over cluster mempunyai fungsi utama menjaga availability. Jika salah satu node anggota cluster mati, sistem harus terus berjalan karena secara otomatis aplikasi akan dipindahkan ke node yang hidup. Proses pemindahan ini biasanya juga ada downtime dalam beberapa detik atau menit tergantung dari konfigurasi server.

3) Global Cluster

Global cluster dibangun menggunakan node yang terkoneksi namun berada di daerah yang berbeda-beda. Walaupun satu cluster, namun node anggota cluster bisa di negara lain, pulau lain bahkan benua lain. Cluster ini sering disebut sebagai disaster recovery. Disaster recovery digunakan untuk menampung data utama di daerah lain agar terhindah dari masalah seperti kebakaran, gangguan listrik maupun bencana alam.

4) Load-balancing Cluster

Cluster ini khusus untuk meningkatkan performa. Dengan load balancing, tingkat skalabilitas cluster menjadi lebih baik.

c) Hal yang Dibutuhkan Cluster

Menurut Jayaswal (2006) ada beberapa komponen yang menjadi kebutuhan untuk membuat sistem cluster:

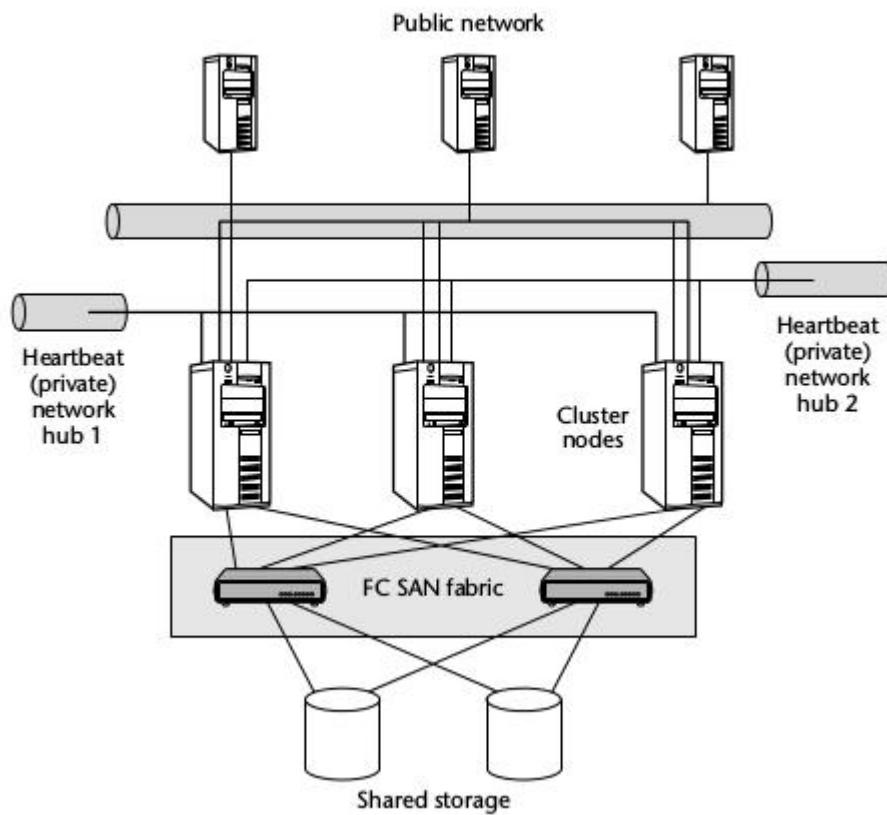
1) Server

Komponen utama cluster adalah server. Server dalam cluster tidak bisa hanya satu sendiri karena sistem cluster harus bisa berjalan meskipun ada server yang mengalami masalah. Server yang digunakan untuk pemrosesan data minimal dua server.

Server di dalam sistem cluster berjalan menggunakan perangkat lunak yang sama, sistem operasi yang sama, dan versi pembaruan software yang sama. Selain software yang sama, hardware juga lebih baik menggunakan spesifikasi yang sama. Hardware mencakup jumlah CPU, jumlah RAM, dan hardware lainnya.

2) Private (heartbeat) Network

Private network merupakan jaringan yang khusus digunakan untuk komunikasi antar server di dalam cluster. Harus dipastikan bahwa jaringan yang bekerja juga memenuhi konsep failover. Switch yang digunakan untuk sistem cluster juga harus redundant sehingga saat salah satu mengalami kendala, switch yang lain akan segera menggantikannya.



Gambar 3. Jaringan Privat Cluster

Sumber : Jayaswal (2006)

3) Administrative (maintenance) network

Administratif network merupakan jaringan yang khusus digunakan untuk pemeliharaan server. Pemeliharaan server bisa mencakup monitoring, konfigurasi dan pekerjaan administratif lainnya.

Saat ingin mengganti IP server, proses penggantian dilakukan melalui remote ke IP administratif kemudian baru dikonfigurasi IP utamanya.

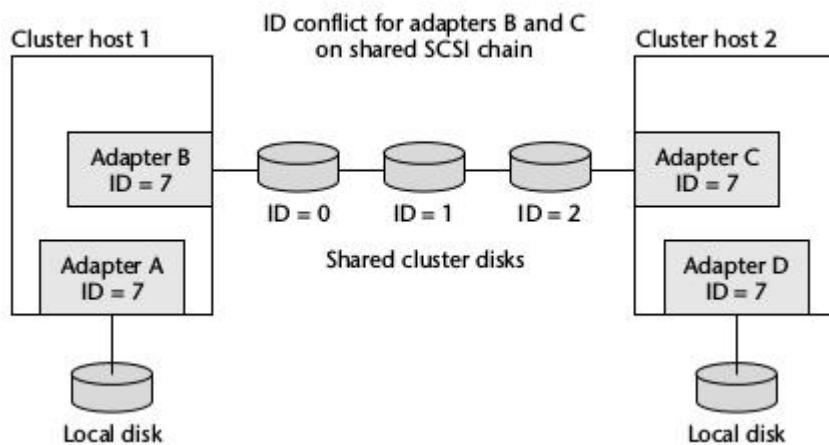
4) Public/Service Network

Public network merupakan jaringan yang terhubung dengan pengguna di luar cluster. Pengguna bisa berupa orang ataupun sistem lain yang juga

menggunakan data dari cluster. Jaringan public bisa diakses melalui sebuah IP public. IP public dalam sistem cluster harus bisa berpindah secara otomatis jika terjadi kesalahan di sistem utama.

5) Shared Disk

Shared Disk merupakan media penyimpanan yang berupa SAN atau NAS. Media penyimpanan ini bertugas sebagai tempat penyimpanan utama untuk data-data yang harus bisa dibaca oleh semua anggota cluster. Sebagai contoh data session, asset gambar ataupun file lainnya. Jika session tidak menggunakan shared disk, maka yang terjadi adalah ketidakkonsistenan sistem cluster.



Gambar 4. Sistem *Shared Disk* Cluster

Sumber : Jayaswal (2006)

6) Local Disk

Local disk merupakan media penyimpanan yang terdapat di masing-masing server. Data yang disimpan bisa berupa sistem operasi, aplikasi binari, file konfigurasi. Data yang ada di local disk juga harus di-mirror menggunakan teknik RAID untuk mengamankan data. Jika ada 2 media penyimpanan yang di-mirror,

maka jika salah satu media penyimpanan mengalami gangguan, server akan tetap berjalan normal.

2. E-Learning

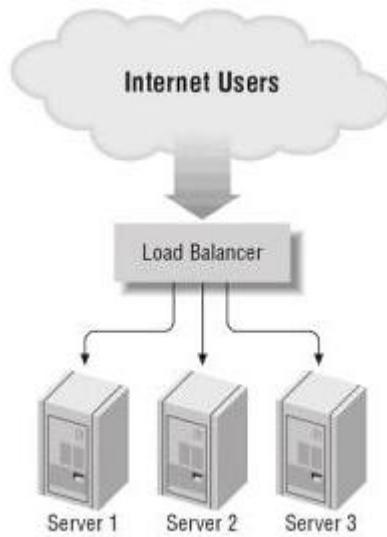
Menurut Muhammad (2011), e-learning tersusun dari dua bagian, yaitu 'e' yang merupakan singkatan dari 'electronica' dan 'learning' yang berarti 'pembelajaran'. Jadi e-learning berarti pembelajaran dengan menggunakan jasa bantuan perangkat elektronika. Jadi dalam pelaksanaannya, e-learning menggunakan jasa audio, video atau perangkat komputer atau kombinasi dari ketiganya ⁹. Dengan kata lain e-learning adalah pembelajaran yang dalam pelaksanaannya didukung oleh jasa teknologi seperti telepon, audio, videotape, transmisi satelite atau komputer.(Tafiardi, 2005) Sejalan dengan itu, Onno W. Purbo (dalam Amin, 2004) menjelaskan bahwa istilah "e" dalam e-learning adalah segala teknologi yang digunakan untuk mendukung usaha-usaha pengajaran lewat teknologi elektronik internet. Internet, satelit, tape audio/video, tv interaktif, dan CD-ROM adalah sebagian dari media elektronik yang digunakan. Pengajaran boleh disampaikan pada waktu yang sama (synchronously) ataupun pada waktu yang berbeda (asynchronously). Secara lebih singkat william Horton mengemukakan bahwa (dalam Sembel, 2004) e-learning merupakan kegiatan pembelajaran berbasis web (yang bisa diakses dari internet). Tidak jauh berbeda dengan itu Brown, 2000 dan Feasey, 2001 (dalam siahaan, 2002) secara sederhana mengatakan bahwa e-learning merupakan kegiatan pembelajaran yang memanfaatkan jaringan (internet, LAN, WAN) sebagai metode penyampaian, interaksi, dan fasilitas yang didukung oleh berbagai bentuk layanan belajar lainnya.

Selain itu, ada yang menjabarkan pengertian e-learning lebih luas lagi. Sebenarnya materi e-learning tidak harus di distribusikan secara on-line baik melalui jaringan lokal maupun intemet. Interaksi dengan menggunakan internetpun bisa dijalankan secara on-line dan real-time ataupun secara off-line atau archieved. Distribusi secara offline menggunakan media CD/DVD pun termasuk pola e-learning. Dalam hal ini aplikasi dan materi belajar dikembangkan sesuai kebutuhan dan di distribusikan melalui media CD/DVD, selanjutnya pembelajar dapat memanfatkan CD/DVD tersebut dan belajar di tempat dimana dia berada (Lukmana,2006).

3. Load Balancer

a) Pengertian Load Balancer

Load balancer merupakan sebuah alat untuk membagi kinerja dari yang sebelumnya satu server menjadi banyak server. Load balancer bisa berupa perangkat lunak maupun perangkat keras. Bourke (2001) menjelaskan load balancing merupakan proses atau teknologi yang mendistribusikan trafik sebuah website menuju ke banyak server menggunakan perangkat berbasis jaringan. Pengakat ini mengintersepsi trafik untuk kemudian mentranslasikan paket data ke beberapa server.



Gambar 5. Arsitektur Load Balancer

Sumber: Jayaswal (2006)

b) Fungsi Load Balancer

Menurut Bourke (2001) ada beberapa fungsi load balancer yaitu

- 1) Melakukan intersepsi terhadap trafik yang berbasis jaringan.
- 2) Memecah banyak trafik menjadi trafik spesifik yang disesuaikan masing-masing konfigurasi kemudian diarahkan ke server tujuan sesuai dengan konfigurasi.
- 3) Memonitor status server tujuan sehingga jika ada salah satu server yang mengalami masalah, load balancer tidak lagi mengarahkan trafik ke server yang bermasalah.
- 4) Menyediakan fitur redundansi untuk load balancer itu sendiri melalui skema fail-over.
- 5) Membaca konten yang akan ditranslasikan, melakukan parsing, membaca URL, membaca cookies, dan parsing XML.

4. Galera Cluster

a) Pengertian Galera Cluster

Galera cluster merupakan perangkat lunak yang berfungsi mengkonversi MySQL biasa menjadi sebuah cluster. MySQL sendiri sebenarnya mempunyai versi cluster namun hanya jika menggunakan versi mysql yang enterprise. Galera dikembangkan secara opensource sehingga banyak modifikasi seperti dikembangkan oleh percona dan mariadb. Menurut Pierre (2014), galera cluster merupakan sebuah solusi yang pertama kali diperkenalkan oleh Codership yang berfungsi untuk membuat database MySQL menjadi multimaster sinkronous.

b) Keuntungan menggunakan galera cluster.

Pierre (2014) menjelaskan banyak keuntungan menggunakan galera cluster, diantaranya sebagai berikut:

1) True multimaster

Galera cluster bisa digunakan untuk menjalankan query write ke semua database server, tidak seperti sistem master-slave yang hanya bisa menulis ke salah satu server.

2) Metode reliplikasi sinkronous

Setiap data menggunakan mode sinkronous untuk melakukan replikasi antar anggota cluster. Metode ini digunakan untuk menyakinkan tidak ada data yang hilang saat sewaktu-waktu database mengalami *crash*.

3) Data konsisten

Semua server anggota cluster mempunyai data yang sama. Tidak ada *lagging* saat melakukan *insert* data sehingga bisa dipastikan data konsisten di semua server anggota cluster.

4) Multithreaded slave

Fitur ini menjadikan galera siap untuk beban pekerjaan yang tinggi. Batas atas galera sesuai dengan spesifikasi server database.

5) Tidak membutuhkan perangkat lunak khusus untuk menejemen HA cluster

Percona bersifat multimaster sehingga tidak perlu menggunakan perangkat lunak untuk failover.

6) Hot Standby

Saat salah satu server mengalami gangguan, tidak perlu adanya konfigurasi ulang yang menyebabkan downtime, fitur ini karena galera berjalan sebagai multimaster database server.

7) Transparent to application

Di sisi pengguna, galera cluster berjalan seperti MySQL biasa. Tidak diperlukan driver khusus untuk melakukan koneksi ke galera cluster

8) WAN

Fitur untuk meningkatkan redundansi dengan penempatan server di daerah lain sebagai cadangan bisa menggunakan WAN.

9) No read and write splitting need

Karena bersifat multimaster, galera cluster bisa digunakan sebagai master maupun slave pada konsep lama. MySQL default hanya menyediakan tools berbayar.

c) Batasan Penggunaan Galera Cluster

Selain keunggulan yang ada di cluster, juga ada batasan jika akan menggunakan cluster, menurut Pierre (2014) berikut ini batasan-batasannya

- 1) Galera cluster hanya mendukung database yang menggunakan *engine* InnoDB.

Jika menggunakan MyISAM, fitur fitur yang ada di galera tidak akan berjalan secara optimal. Ada pula database TokuDB yang sedang dikembangkan agar bisa berjalan diatas galera cluster.

- 2) Semua tabel harus mempunyai primary keys.

Primary keys di semua table digunakan untuk menghindari perbedaan antrian eksekusi query diantara semua anggota galera cluster.

- 3) Tidak mendukung *locking* dan *unlocking* tabel.

Secara standar, galera sudah mengimplementasi metode locking tersendiri sehingga jika pengguna menggunakan fitur yang sama, pasti akan terjadi tabrakan penggunaan tabel locking maupun unlocking.

- 4) Galera cluster menonaktifkan query cache

Query cache hanya bisa berjalan di sistem MySQL biasa namun tidak bisa berjalan di Galera Cluster.

- 5) Query log harus disimpan dalam bentuk file.

Query log secara standar bisa disimpan di tabek ataupun di file. Jika menggunakan galera cluster, hanya tersedia opsi penyimpanan di file.

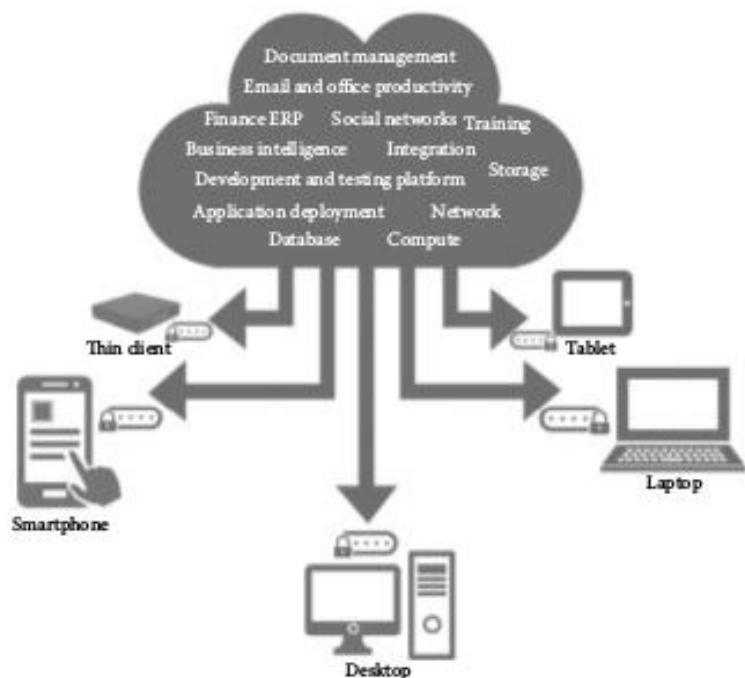
5. Cloud Computing

- a) Pengertian Cloud Computing

Sebelum era cloud computing, setiap individu yang membutuhkan komputasi harus berinvestasi biaya untuk membeli hardware, software , perangkat jaringan, dan storage. Investasi ini biasanya membutuhkan banyak biaya tambahan seperti biaya perawatan dan biaya operasional seperti biaya listrik. Menurut

Chandrasekaran (2015), biaya bisa lebih besar jika digunakan untuk perusahaan daripada untuk penggunaan individu. Perusahaan harus menggunakan listrik dari beberapa sumber untuk memastikan sistem berjalan tanpa kendala.

Cloud computing sangat ekonomis dan aman, dengan cloud computing juga jika kita kehilangan data di komputer pribadi kita, data masih aman karena tersimpan secara *remote* di server cloud (Chandrasekaran:2015).



Gambar 6. Cloud Computing

Sumber: Chandrasekaran (2015)

Pengertian sederhana dari cloud computing adalah menyimpan, mengakses, dan mengolah data menggunakan sistem yang berjalan *remote*, tidak dilakukan dengan komputer lokal yang menggunakan komputer personal pengguna cloud

computing (Chandrasekaran:2015). Cloud computing membutuhkan internet untuk mengirimkan data-data kepada penggunanya.

b) Jenis-jenis Cloud Computing

Badger dkk (2012) menjelaskan jenis jenis cloud computing menjadi beberapa jenis yaitu

User Application	IaaS Model	PaaS Model	SaaS Model
Application	Application	Application	Application
Data	Data	Data	Data
Runtime (Libraries)	Runtime (Libraries)	Runtime (Libraries)	Runtime (Libraries)
Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization
Server	Server	Server	Server
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Gambar 7. Jenis-jenis Cloud Computing

Sumber: Badger dkk (2012)

1) Software as s Service (SaaS)

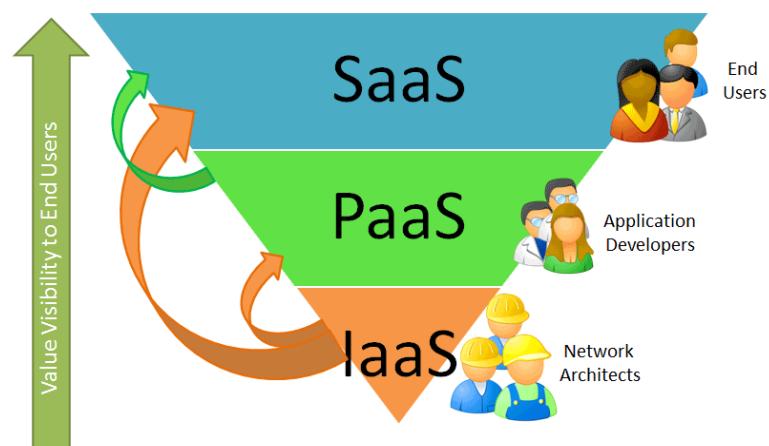
Teknologi cloud SaaS memungkinkan pengguna untuk menggunakan aplikasi yang disediakan oleh penyedia layanan saas yang berjalan diatas infrastruktur cloud. Pengguna tidak perlu untuk memanajemen dan kontrol terhadap infrastruktur cloud seperti server fisik, jaringan, sistem operasi, dan storage.

2) Platform as a Service (PaaS)

Teknologi ini memungkinkan pengguna untuk memakai atau mengunggah sendiri aplikasi yang telah dibuat menggunakan bahasa pemrograman tertentung yang bisa berjalan di server penyedia PaaS. Pengguna tidak perlu memanajemen server fisik, jaringan, sistem operasi, dan storage namun tetap harus memanajemen aplikasi mereka seperti penyesuaian pembaharuan perangkat lunak, *patch* dan penambahan fitur aplikasi.

3) Infrastruktur as a Service (IaaS)

IaaS memungkinkan pengguna untuk mengkonfigurasi sendiri spesifikasi server yang akan digunakan oleh aplikasi. Pengguna bisa mengkonfigurasi jumlah server, jumlah RAM, CPU, maupun terkait jaringan internet. Pengguna harus memilih sendiri sistem operasi dan melakukan manajemen secara keseluruhan terhadap server tersebut. Semua server berjalan secara virtual pengguna yang mengelola, namun karena server fisik dari pihak penyedia layanan, maka server fisik dikelola oleh penyedia layanan.



Gambar 8. Jenis Cloud Berdasar Pengguna

Sumber : Qarea

c) Karakteristik Cloud Computing

Chandrasekaran (2015) menjelaskan lima karakteristik cloud computing sebagai berikut:

1) On-demand self-service

Pengguna bisa mengubah pemesanan, membuat pemesanan produk sesuai dengan kebutuhan sendiri. Pengubahan pemesanan seharunya bisa langsung dilakukan oleh pengguna sendiri tanpa harus ada interaksi dari penyedia layanan cloud.

2) Broad network access

Kemampuan jaringan cloud untuk bisa diakses dari alat apapun. Alat yang dimaksud adalah server diluar cloud yang mengakses cloud tersebut maupun perangkat mobile seperti telepon selular, tepelon pintar, komputer jinjing, maupun perangkat *portable* lainnya.

3) Elastic resource pooling

Penyedia layanan cloud seharunya bisa memastikan bahwa pengguna bisa menaikkan dan mengurangi kapasitas komputasi secara fleksibel. Kapasitas komputasi bisa berupa storage, RAM, CPU amupun bandwith jaringan. Walaupun pengguna tidak mengetahui kapasitas total dari penyedia layanan, namun saat pengguna akan menambah kapasitas yang dipakai, maka penyedia cloud bisa mengakomodasi kebutuhan tersebut.

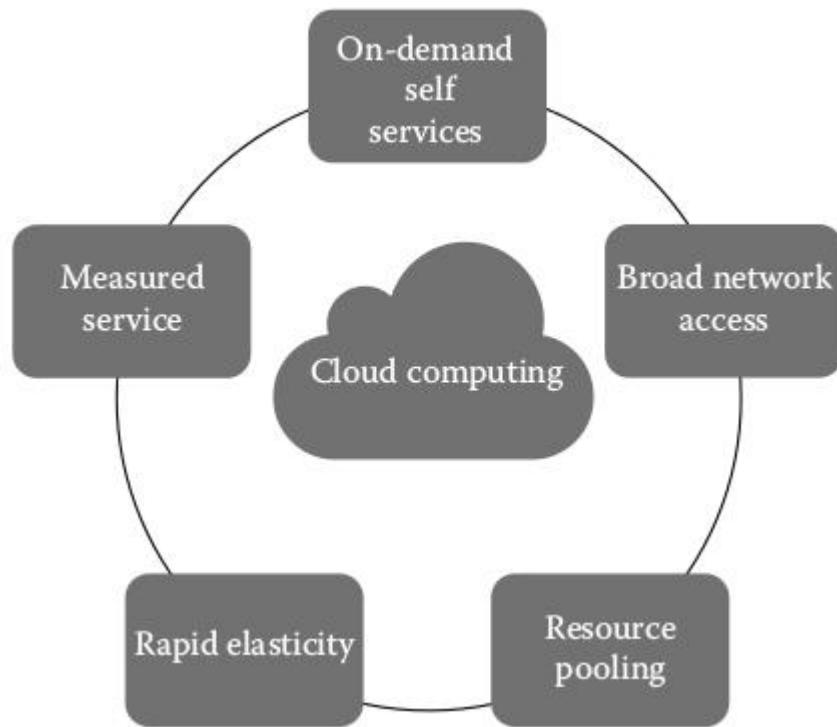
4) Rapid elasticity

Tidak perlu waktu yang lama untuk penyesuaikan skalabilitas. Cloud harus mempunyai tingkat skalabilitas tinggi sehingga saat pengguna membutuhkan

perubahan konfigurasi dan kapasitas, layanan cloud bisa tetap melayani kebutuhan tersebut.

5) Measured service

Segala bentuk penggunaan sumber daya cloud harus bisa terukur dengan sangat detail. Cloud biasanya bersifat pay-as-you-go yang artinya pengguna hanya membayar apa yang digunakan saja. Untuk bisa menentukan berapa biaya yang dikeluarkan pengguna, semua sumber daya yang dipakai harus dihitung seperti kapasitas CPU, RAM, storage, maupun koneksi internet server.



Gambar 9. Karakteristik Cloud Computing

Sumber: Chandrasekaran (2015)

6. Tahap Pengembangan Arsitekturn Infrastruktur

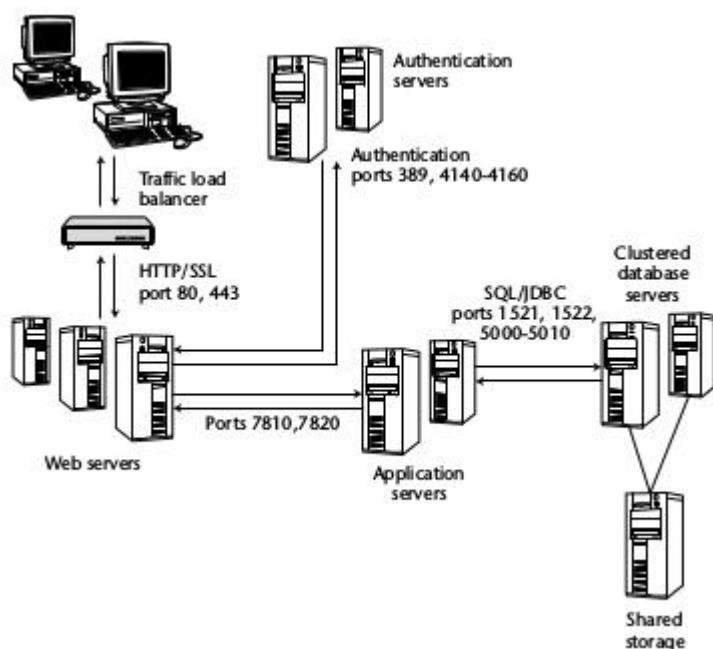
Jayaswal(2006) dalam bukunya "Administering Data Centers: Servers, Storage, and Voice over IP" menjelaskan ada beberapa tahap untuk melakukan konsolidasi arsitektur, tahap-tahapnya adalah sebagai berikut:

a) Tahap Analisis dan Dokumentasi Arsitektur Lama

Tahap pertama merupakan tahap yang paling penting . dalam tahap ini semua bagian infrastruktur yang lama dikumpulkan informasinya dan dibuat sebuah dokumentasi. Ada 5 hal yang harus digali informasinya, yaitu:

1) Analisis Kebutuhan Aplikasi

Setiap aplikasi yang dijalankan diatas sebuah infrastruktur harus dianalisis. Analisis bisa mencakup data-flow, melakukan interview dengan pengembang, arsitek sebelumnya maupun sistem administrator.



Gambar 10. Contoh Analisis Kebutuhan Aplikasi

Sumber: Jayaswal (2006)

2) Analisis Kebutuhan Hardware

Setelah semua kebutuhan aplikasi dikumpulkan, analisis kebutuhan hardware menjadi langkah selanjutnya. Analisis mencakup varifikasi inventori hardware dan memperkirakan penggunaan sumber daya hardware. Inventori hardware mencakup tentang berapa server yang ada, berapa perangkat jaringan yang ada, termasuk jumlah CPU, RAM, dan periperal lainnya.

3) Analisis Kebutuhan Media Penyimpanan

Ada 3 aspek yang harus dianalisis dalam analisis media penyimpanan. Aspek yang pertama adalah perangkat keras storage, aspek ini menghitung berapa total media penyimpanan yang dipunyai, apakah menggunakan RAID, dan apakah ada storage yang ada di luar daerah sebagai disaster-recovery. Aspek kedua adalah konfigurasi logik. Konfigurasi logik dalam media penyimpanan menganalisa jenis konfigurasi logik yang digunakan dalam media penyimpanan. Contoh konfigurasi logiknya adalah, apakah menggunakan LVM, dan partisi jenis apa yang dipakai.

4) Analisis Kebutuhan Jaringan

Menganalisis kebutuhan jaringan di arsitektur lama, dianalisis tentang performa jaringan dan seperti apa topologi jaringan yang saat ini sudah berjalan.

5) Analisis Kebutuhan Operasional

Analisis kebutuhan operasional mencakup analisis tenaga ahli, analisis standar operasional. Hal ini penting karena berdasarkan penelitian 80 persen kasus gangguan datacenter adalah karena kesalahan manusia dan standar operasional yang kurang baik.

b) Mendesain Arsitektur Baru

1) Analisis Informasi Arsitektur Lama

Setelah proses analisis arsitektur lama, data data yang terkumpul harus dianalisis agar kemudian dijadikan acuan untuk mengembangkan arsitektur yang baru.

2) Analisis Kebutuhan Arsitektur Baru

Setelah melihat kelemahan yang ada di arsitektur lama, tahap selanjurnya adalah membuat detail rancangan penyelesaian masalah yang ada di arsitektur lama dan membuat daftar kebutuhan apa saja yang dibutuhkan oleh arsitektur yang baru.

3) Membuat Desain Arsitektur Dasar

Tahap ini terdiri dari 3 desain yang harus dibuat. Desain pertama adalah desain arsitektur server. Arsitektur server yang baru harus mengakomodasi semua penyelesaian masalah. Selain mengelola hardware, tahap ini juga akan menentukan apakah akan menggunakan sistem server biasa, grid computing, cluster, paralel computing ataupun jenis jenis arsitektur lainnya.

Tahap kedua adalah desain storage. Desain storage dimaksudkan untuk menentukan metode apa saja yang akan digunakan di arsitektur yang baru, apakah menggunakan SAN, NAS, atau media penyimpanan lokal.

Tahap ketiga adalah desain arsitektur jaringan. Tahap ini menentukan tipe jaringan yang dipakai, membagi jaringan berdasar fungsinya seperti apakah untuk backup, produksi, ataupun untuk NAS. Masing-masing fungsi tersebut mempunyai kebutuhan network yang berbeda-beda.

4) Mengukur Kembali Semua Kebutuhan

 Semua yang sudah didesain harus bisa memenuhi standar minimal performa yang diperlukan oleh aplikasi.

5) Membuat Prototype Arsitektur

 Setelah semua selesai dialisis dan dihitung, tahap selanjutnya adalah melakukan prototype arsitektur yang baru. Semua komponen arsitektur dibangun namun dengan skala yang kecil.

6) Ujicoba Prototype

 Setelah prototype selesai dibuat, arsitektur yang baru harus segera diujicoba. Ada 3 aspek yang harus diujicoba. Aspek pertama adalah *setup*, ujicoba aspek ini untuk memastika semua server, storage, jaringan dan aplikasi berjalan normal sama seperti saat berjalan di arsitektur lama.

 Aspek yang kedua adalah aspek keamanan, diuji apakah semua port dan firewall sudah dikonfigurasi secara benar. Aspek yang terakhir adalah aspek fungsionalitas. Aspek fungsionalitas mencakup kerja aplikasi, ujicoba backup, monitoring.

7) Revisi Prototype Arsitektur Baru

 Jika saat ujicoba terdapat kendala, maka tahap selanjutnya adalah melakukan revisi desain arsitektur.

8) Dokumentasi Prototype Arsitektur Baru

 Setelah uji coba selesai, segala hal dalam ujicoba harus di dokumentasikan agar bisa segera diimplementasikan di infrastruktur yang baru. Dokumentasi yang dibuat biasanya berisi tentang asesmen dari data yang sudah diperoleh,

kebutuhan arsitektur baru, arsitektur yang baru, hasil ujicoba dan revisi jika ada dilampirkan revisi desain arsitektur setelah uji coba.

c) Membangun Arsitektur Menggunakan Desain Arsitektur Baru

1) Membangun Arsitektur Baru

Dalam tahap ini dokumentasi dan desain arsitektur yang sudah diuji coba dalam tahap prototype akan diimplementasikan. Semua komponen disesuaikan dengan kebutuhan nyata aplikasi yang akan berjalan diatas arsitektur yang baru.

2) Mempersiapkan Migrasi Data

Mempersiapkan media apa yang akan digunakan untuk migrasi data dari infrastruktur lama ke infrastruktur baru. Ada banyak pilihan metode migrasi, bisa melalui FTP, SCP, atau menggunakan tape backup. Selain file aplikasi, untuk database juga harus disiapkan metode migrasinya apakah hanya perlu export atau perlu dengan metode replikasi.

3) Backup Data dari Arsitektur Lama

Proses backup ini bukan proses backup untuk kemudian dilakukan restore di infrastruktur baru, namun untuk keamanan data jika terjadi sesuatu saat proses migrasi.

4) Migrasi Data dari Arsitektur Lama ke Arsitektur Baru

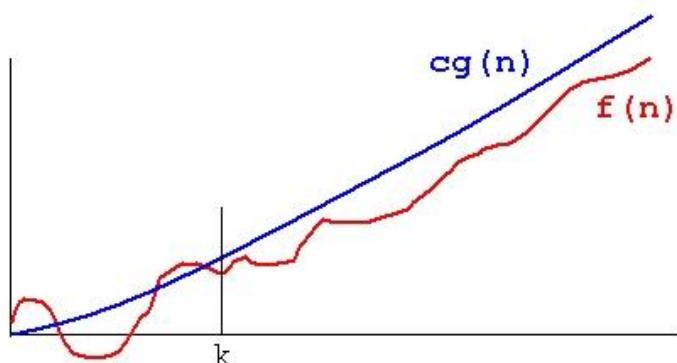
Migrasi data dilakukan dengan alat migrasi yang sudah ditentukan sebelumnya, setelah proses migrasi data selesai, biasanya dilanjutkan dengan mengubah data DNS jika berupa web. Setelah proses migrasi selesai, sistem yang baru harus di tes kembali untuk memastikan sistem berjalan normal.

7. Httperf

Httperf merupakan perangkat lunak yang dikembangkan oleh Hewlett-Packard pada tahun 1998. Httperf menurut Jin (1998) memiliki fungsi untuk mengukur performa server dengan cara memberikan beban kepada server. Httperf dibuat menggunakan bahasa pemrograman C.

8. Notasi Big-O

Notasi Big-O adalah sebuah cara pengukuran kinerja sebuah algoritma. Notasi Big-O digunakan untuk mengukur efisiensi dari waktu eksekusi ataupun penggunaan memori dari sebuah algoritma. Paul (2005) menjelaskan bahwa dengan notasi Big-O, pembuat algoritma bisa memperkirakan bagaimana kinerja algoritma yang dibuat, bahkan bisa memperkirakan penggunaan *resource* jika ada banyak algoritma yang dipakai. Berdasar notasi Big-O setiap penambahan algoritma, maka semakin besar juga waktu yang dibutuhkan dan *resource* yang digunakan.



Dalam notasi Big-O terdapat algoritma linear dengan model sebagai berikut:

$$T(n) = O(n)$$

Algoritma linear menunjukkan bahwa waktu eksekusinya sebanding dengan jumlah data, dan algoritma linear saat ini menjadi yang paling banyak diimplementasi karena merupakan kondisi optimal dalam membuat algoritma.

9. Variabel Performa Server Berdasar ITU-T G.1030 11/2005

a) Throughput

Throughput adalah jumlah bit yang diterima dengan sukses perdetik melalui sebuah sistem atau media komunikasi dalam selang waktu pengamatan tertentu. Umumnya throughput direpresentasikan dalam satuan bit per second (bps). Aspek utama throughput yaitu berkisar pada ketersediaan bandwidth yang cukup untuk menjalankan aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh suatu aplikasi saat melewati jaringan. Aspek penting lainnya adalah error (pada umumnya berhubungan dengan link error rate) dan losses (pada umumnya berhubungan dengan kapasitas buffer).

b) Response time

Response time adalah selang waktu antara user memasukkan suatu perintah ke dalam sistem hingga mendapat balasan selengkapnya dari sistem. Pada ITU-T G.1030 11/2005 (2005:4), dijelaskan bahwa response time memiliki range yang dapat dibagi ke dalam beberapa kategori:

1) Instantaneous experience (0,1 detik)

0,1 detik adalah perkiraan batasan waktu seorang pengguna merasakan bahwa sistem langsung bereaksi (instan) setelah memasukkan input atau perintah ke sistem, contohnya pada chatting.

2) Uninterrupted experience (1,0 detik)

1,0 detik adalah perkiraan batasan waktu seorang pengguna tetap tidak terganggu, meskipun pengguna mulai merasa bahwa sistem tidak merespon input secara instan, contohnya pada gaming.

3) Loss of attention (10 detik)

10 detik adalah perkiraan batasan waktu seorang pengguna tetap fokus memperhatikan dialog. Pada delay yang lebih lama, pengguna umumnya akan melakukan pekerjaan lain sembari menunggu komputer selesai merespon, sehingga pengguna harus diberikan feedback bahwa komputer akan segera memberikan respon.

c) Reply Connection Client

Merupakan jumlah request yang diterima oleh server, kemudian server memberikan pengiriman pemberitahuan kepada client bahwa request telah diterima. Selain berpengaruh terhadap latency, reply connection juga berpengaruh terhadap tinggi throughput. Dalam praktiknya, throughput merupakan rata-rata besarnya content (jumlah bit) yang diterima dikalikan jumlah permintaan content yang dibalas (replied connection), kemudian dibagi dengan waktu pengukuran (test durations).

d) Error Connection Client

Merupakan jumlah request dari client yang ditolak atau tidak mampu terjawab oleh server. Jika terdapat 10 permintaan koneksi ke server dan reply connection-nya ada 8, maka error connection-nya sama dengan 2. Waktu pengukuran (test durations) secara tidak langsung dipengaruhi oleh response

time masing-masing permintaan. Semakin tinggi response time masing-masing permintaan, maka akan semakin lama pengukuran berlangsung. Response time tidak berpengaruh secara langsung terhadap waktu pengukuran karena waktu pengukuran juga dipengaruhi oleh jumlah permintaan dan jumlah thread.

B. Kajian Penelitian yang Relevan

1. Hasil penelitian dari Omar Muhammad Altoumi Alsyabani yang berjudul "Performa Algoritma Load Balance pada Server Web Apache dan Nginx dengan Database PostgreSQL". Tujuan dari penelitian ini adalah mengetahui kinerja web load balancer. Masing-masing algoritma load balance digunakan pada Apache dan Nginx. Database yang digunakan adalah PostgreSQL.
2. Hasil penelitian tesis dari Didik Ariwibowo yang berjudul "Optimalisasi Cluster Server LMS dan IPTV dengan Variasi Algoritma Penjadwalan". Penelitian ini bertujuan untuk menerapkan sistem cluster server LMS dan server IPTV dengan berbagai macam algoritma penjadwalan kemudian juga untuk mendapatkan algoritma penjadwalan yang terbaik pada sistem cluster yang sudah dibangun.

C. Kerangka Berpikir

Penelitian ini berawal karena ada masalah rendahnya skalabilitas untuk e-learning Moodle sehingga diperlukan dibuat sebuah cluster. Pengembangan cluster untuk web server sudah ditemukan, namun belum ada standar dari moodle untuk mengembangkan database server menggunakan sistem cluster. Ada beberapa opsi untuk implementasi sistem database cluster untuk Moodle diantaranya dengan Galera Cluster. Selain bertujuan membuat arsitektur database cluster untuk Moodle, penelitian ini juga membandingkan kinerja

database load balancer antara HAProxy dan MaxScale. Pengembangan arsitektur dan implementasi arsitektur database cluster menggunakan tahapan yang dijelaskan oleh Jayaswal(2006) dalam bukunya "Administering Data Centers: Servers, Storage, and Voice over IP". Untuk membandingkan kinerja dua load balancer, dilakukan dengan cara pengambilan data menggunakan perangkat lunak httpperf dengan aspek aspek yang dihitung disesuaikan dengan standar ITU-T G.1030 11/2005. Ada 4 aspek yang dihitung yaitu throughput, response time, reply client dan error client. Pengujian perbedaan menggunakan metode statistika Uji Beda(Uji T).

BAB III

METODE PENELITIAN

A. Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini adalah penelitian deskriptif komparatif dengan pendekatan kuantitatif. Pengertian deskriptif menurut (Nazir, 2005) adalah suatu metode dalam meneliti status sekelompok manusia, suatu objek, suatu set kondisi, suatu sistem pemikiran, ataupun suatu kelas peristiwa pada masa sekarang. Tujuan dari penelitian deskriptif adalah untuk membuat deskripsi, gambaran, atau lukisan secara sistematis, faktual dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antar fenomena yang diselidiki. Dalam metode deskriptif peneliti bisa membandingkan fenomena-fenomena tertentu sehingga merupakan suatu studi komparatif.

Penelitian komparatif adalah penelitian yang membandingkan keberadaan satu variabel atau lebih pada dua atau lebih sampel yang berbeda, atau pada waktu yang berbeda (Sugiyono, 2013).

B. Metode Penelitian

Penelitian ini menggunakan penelitian deskriptif yaitu suatu metode dalam meneliti status sekelompok manusia, suatu obyek, suatu kondisi, suatu sistem pemikiran ataupun suatu kelas peristiwa pada masa sekarang. Tujuan dari penelitian deskriptif ini adalah untuk membuat deskripsi, gambaran atau lukisan secara sistematis, faktual dan akurat mengenai fakta-fakta, sifat-sifat antar fenomena yang diteliti.

Metode yang digunakan dalam penelitian kuantitatif ini adalah metode observasi terstruktur. Penggunaan metode dalam penelitian disesuaikan dengan masalah serta tujuan penelitian tersebut. Oleh sebab itu, metode penelitian sangat penting dalam pelaksanaan, pengumpulan dan analisis data.

C. Prosedur Penelitian

Dalam menyelesaikan penelitian ini prosedur penelitian yang digunakan adalah sebagai berikut:

1. Mencari ide atau gagasan penelitian

Ide atau gagasan penelitian muncul karena MaxScale menambahkan beberapa fitur yang sangat berguna untuk meningkatkan aspek skalabilitas, namun dalam observasi sebelumnya menggunakan website drupal, terjadi penurunan kecepatan saat *load balancer* diubah dari HAProxy menjadi MaxScale.

2. Melakukan study literature

Sebelum melakukan penelitian, peneliti melakukan studi pendahuluan yang dilakukan dengan mencari informasi dan referensi yang terkait untuk mendukung penelitian.

3. Menentukan rumusan masalah

Setelah melakukan studi literature maka menentukan rumusan masalah yang tepat.

4. Menentukan tujuan penelitian

Menentukan tujuan- tujuan yang akan dicapai dalam kegiatan penelitian agar tidak menyimpang dari permasalahan yang telah dirumuskan.

5. Melakukan pengambilan data

Data diambil dengan observasi terhadap performa kecepatan website moodle saat load balancer diganti dari HAProxy ke MaxScale.

6. Menganalisis data

Data yang telah dikumpulkan diolah lebih lanjut kemudian disajikan dalam bentuk statistic dan selanjutnya dianalisis.

7. Merumuskan kesimpulan

Hasil analisis data akan memberikan kesimpulan penelitian yang merupakan kegiatan akhir penelitian

D. Lokasi Penelitian

Penelitian dilakukan di Public Cloud Amazon Web Service (AWS). AWS digunakan untuk melakukan instalasi semua server untuk ujicoba penelitian. Datacenter yang dipilih adalah datacenter Singapura karena mempertimbangkan *latency*.

Penelitian dilakukan di sistem clud Amazon Web Service menggunakan layanan EC2. Untuk memastikan tidak ada gangguan dari luar, dalam penelitian ini menggunakan Provisioned IOPS. Provisioned IOPS memastikan resource yang digunakan hanya dipakai oleh komputer peneliti dan tidak dibagi dengan pengguna AWS yang lain.

Agar ketika benchmark tidak terjadi intervensi jaringan dari luar lingkungan penelitian, maka dalam penelitian ini menggunakan server benchmark yang lokasinya ada dalam satu jaringan datacenter AWS.

E. Populasi dan Sampel Penelitian

1. Populasi Penelitian

Sebuah penelitian selalu berkaitan dengan kegiatan mengumpulkan dan menganalisis suatu data, menentukan populasi merupakan langkah yang penting. Populasi penelitian merupakan kelompok keseluruhan orang, peristiwa atau sesuatu yang ingin diselidiki oleh peneliti. Sugiyono (2013), menjelaskan bahwa: "Populasi adalah wilayah generalisasi yang terdiri dari objek atau subjek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya."

Berdasarkan penjelasan tersebut, maka penelitian ini populasinya adalah website e-learning moodle yang menggunakan spesifikasi dan arsitektur yang sama dengan penelitian ini.

2. Sampel penelitian

Menurut Sugiyono (2013), sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi. Sampel uji penelitian ini mengambil sampel dengan teknik purposive sampling. Menurut Juanda (2014), untuk menentukan jumlah sampel yang dibutuhkan, digunakan rumus *cross sectional* dengan besar populasi tidak diketahui. Untuk menentukan jumlah sampel yang dibutuhkan, digunakan rumus cross sectional sebagai berikut:

$$n = \frac{Z^2 p(1 - p)}{d^2}$$

Keterangan :

n = jumlah sampel minimal yang diperlukan

Z = score Z , berdasarkan nilai α yang diinginkan

α = derajat kepercayaan

d = toleransi kesalahan

p = proporsi kasus yang diteliti dalam populasi, jika p tidak diketahui maka gunakan p terbesar. p terbesar yaitu $p = 0.5$

$1-p = q$, yaitu proporsi untuk terjadinya suatu kejadian. Jika penelitian ini menggunakan p terbesar, maka $q = 1-p = 1=0.5$

Pada penelitian ini, ditentukan bahwa batas toleransi kesalahan adalah $5\% = 0.05$.

A	$1 - \alpha$	$Z_1 - \alpha/2$	$Z_1 - \alpha$
1%	99%	2.58	2.33
5%	95%	1.96	1.64
10%	90%	1.64	1.28

Sesuai dengan rumus cross sectional dimana $Z_1 - \alpha/2$, maka besaran score Z yang akan diambil adalah sesuai dengan kolom ketiga. Pada penelitian ini, derajat kepercayaan yang digunakan adalah 5%, maka $Z_1 - \alpha/2 = 1.96$. Jika sudah ditetapkan bawah score $Z = 1.96$, maka $Z^2 = 3.84$ atau dibulatkan

menjadi 4. Rumus untuk mengetahui jumlah sampel yang dibutuhkan sering kali dituliskan sebagai berikut:

$$n = \frac{4 p q}{d^2}$$

Maka, dari rumus di atas jumlah sampel yang dibutuhkan pada penelitian ini adalah 384. Menurut Kusnandarr (2014), pada umumnya distribusi sampling dari rata - rata sampel akan mendekati distribusi Normal jika ukuran sampelnya lebih besar dari 30. Sebuah studi simulasi Monte Carlo oleh Smith dan Wells menunjukkan bahwa ukuran sampel 15 sudah sesuai dengan teorema limit pusat untuk Distribusi Normal, tetapi untuk Distribusi Bimodal pada saat ukuran sampel 30 baru sesuai dengan teorema limit pusat. Dengan data tersebut, distribusi sampling dalam penelitian ini sudah normal karena lebih dari 30.

F. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah observasi terstruktur. Sugiyono (2013) mengungkapkan bahwa observasi tidak terbatas pada orang, tetapi juga objek-objek alam yang lain. Teknik pengumpulan data dengan observasi digunakan bila penelitian berkenaan dengan perilaku manusia, proses kerja, gejala-gejala alam dan bila responden yang diamati tidak terlalu besar.

G. Instrumen Penelitian

Instrumen penelitian yang digunakan dalam penelitian ini adalah perangkat lunak httpperf. Httpperf merupakan perangkat lunak yang dirilis oleh perusahaan

HP (*Hawlett Packard*) untuk menghitung performa sebuah website. Menurut Orsini(2007), Htperf merupakan alat yang digunakan untuk memberikan berbagai jenis beban untuk HTTP server. Htperf bisa mengukur performa server setelah diberikan beban tertentu. Instrumen pengukur kecepatan website diukur menggunakan standar internasional ITU-T G.1030 11/2005. Variabel yang diteliti di penelitian ini adalah Troughput, Response Time, Reply Client dan Error Client.

H. Teknik Analisis Data

Analisis data dilakukan dengan menggunakan uji statistik yang disebut Uji Beda. Uji statistik yang digunakan dalam penelitian ini adalah : Uji parametrik untuk dua sampel tidak berhubungan (*two independent samples*).

Analisis menggunakan perangkat lunak Microsoft Excel dengan tambahan Add-on Alalysys Toolpack. Data yang dianalisis adalah data throughput, response time, reply client dan error client yang diperolah dari observasi.

BAB IV

HASIL DAN PEMBAHASAN

A. Tahap Analisis dan Dokumentasi Arsitektur Lama

1. Analisis Kebutuhan Aplikasi

Aplikasi yang dibangun dalam peneliti adalah e-learning moodle. Kebutuhan dasar untuk menjalankan moodle adalah menggunakan Apache, PostgreSQL/MySQL/MariaDB dan PHP. Moodle bisa diinstall di sistem operasi Linux maupun Windows.

2. Analisis Kebutuhan Hardware

Berikut spesifikasi hardware yang dibutuhkan untuk menjalankan aplikasi Moodle:

- a) Processor: 1GHz (minimal), disarankan 2GHz dual core atau lebih.
- b) Memory: 256MB (Minimal), sangat disarankan 1GB atau lebih.

3. Analisis Kebutuhan Media Penyimpanan

Kebutuhan media penyimpanan terbagi menjadi dua jenis yaitu, media penyimpanan utama dan untuk backup. Berikut detail kebutuhan media penyimpanan Moodle:

- a) Kapasitas media penyimpanan utama : 160MB (minimal), disarankan diatas 5GB namun disesuaikan dengan kebutuhan.
- b) Backup: Minimal sama dengan kapasitas media penyimpanan utama namun bisa lebih besar untuk menampung lebih banyak backup berdasarkan tanggal operasi.

4. Analisis Kebutuhan Jaringan

Kapasitas jaringan disesuaikan dengan jumlah pengguna yang mengakses moodle secara bersamaan. Server yang ditempatkan di datacenter biasanya mempunyai kecepatan jaringan sebesar 100Mbps dan 1Gbps. Kecepatan jaringan sangat ditentukan oleh penyedia layanan network yang dipakai, baik untuk koneksi lokal maupun internet.

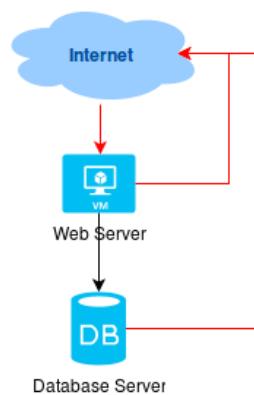
5. Analisis Kebutuhan Operasional

Setelah selesai diinstall, Moodle harus dimonitor kondisinya, untuk memonitor kondisi server menggunakan aplikasi internal seperti *top*, *htop*, *iostat*, *iotop*, dan *iftop* dan menggunakan aplikasi monitoring tambahan new relic.

B. Mendesain Arsitektur Baru

1. Analisis Data Arsitektur Lama

Arsitektur Moodle yang lama merupakan arsitektur minimal untuk Moodle yaitu terdiri dari 1 Web Server dan 1 Database Server. Arsitektur ini bisa menjalankan Moodle secara optimal namun tingkat skalabilitasnya rendah.



Gambar 11. Arsitektur Lama Moodle

Gambar diatas menunjukkan bahwa minimal moodle mempunyai dua komponen yaitu web server dan database server. Web server berfungsi untuk melakukan pengolahan data sedangkan database server digunakan untuk penyimpanan data. Spesifikasi web server dan database server seharusnya sama baiknya, namun dalam kasus tertentu terkadang terjadi ketidakseimbangan beban sehingga perlu dioptimasi lebih lanjut seperti dengan menambahkan web server tambahan maupun meningkatkan melalui mekanisme *upgrade* server.

2. Analisis Kebutuhan Arsitektur Baru

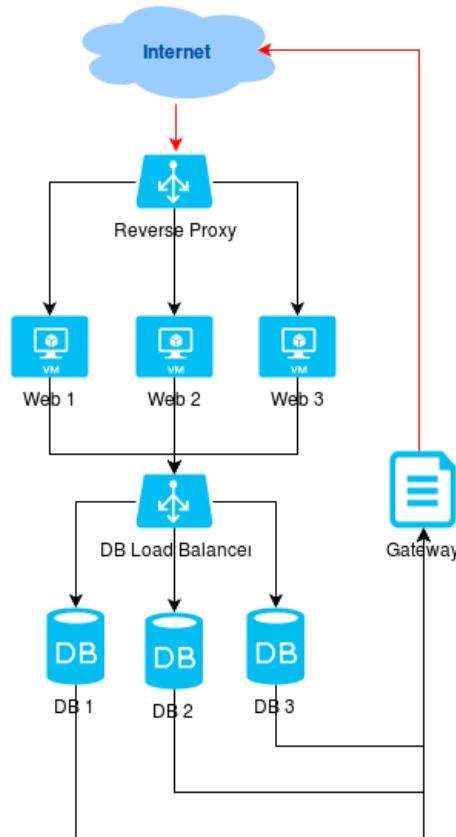
Arsitektur moodle yang akan diteliti menggunakan konsep cluster. Terdapat beberapa web server, database server, web load balancer dan database load balancer. Arsitektur yang baru didesain agar lebih *scalable* oleh karena itu menggunakan load balancer sebagai pembagi beban untuk masing-masing cluster yaitu web cluster dan database cluster.

Arsitektur Moodle yang baru diinstall di Amazon Web Services (AWS), AWS merupakan sebuah perusahaan penyedia layanan cloud yang pusatnya ada di Amerika Serikat. Walauun berpusat di Amerika Serikat, namun AWS mempunyai datacenter yang tersebar di seluruh dunia. Mendesain arsitektur yang menggunakan AWS berbeda dengan mendesain arsitektur datacenter seperti biasanya. Mendesain arsitektur di AWS kita tidak perlu memikirkan tentang pemasangan server fisik, router, switch maupun cabling karena AWS sudah membantu memanajemen hal-hal tersebut dan pengguna hanya perlu membuat server secara virtual melalui panel kontrol yang sudah disediakan oleh AWS. Sistem yang digunakan AWS dinamakan sistem server cloud.

Ada banyak penyedia cloud selain AWS namun peneliti dalam hal ini menggunakan AWS karena pertimbangan sistem di AWS yang stabil, kemudahan manajemen server dan sistem pembayaran yang sesuai kebutuhan.

3. Membuat Desain Arsitektur Dasar

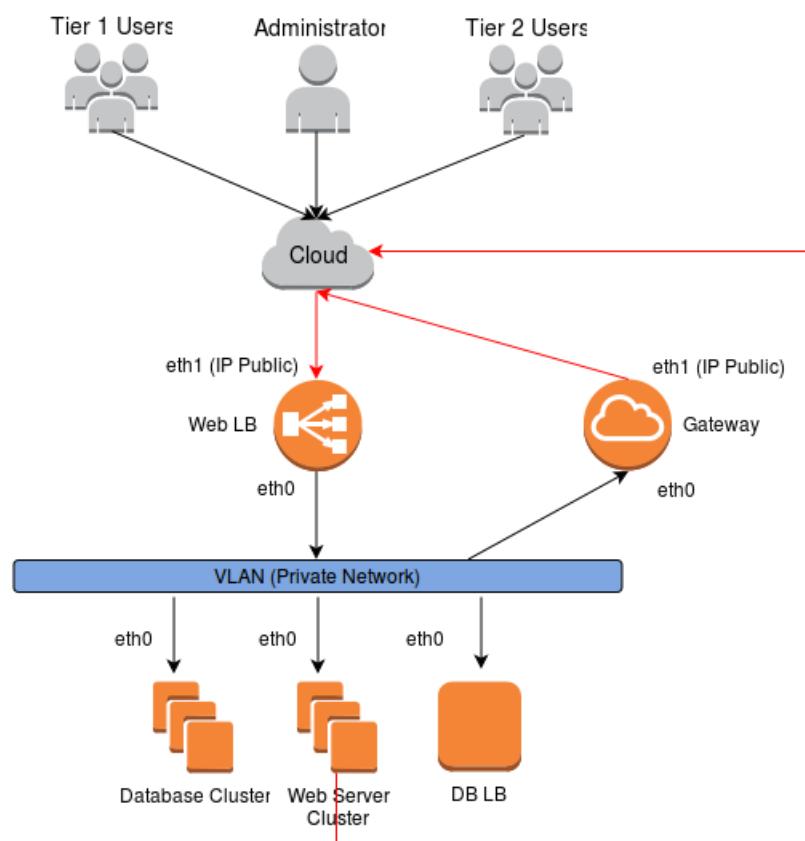
Desain arsitektur dasar yaitu desain arsitektur yang mencakup kebutuhan dasar sebuah arsitektur yakni web load balancer, web server, database server dan load balancer. Pada tahap ini arsitektur dibuat seideal mungkin. Dalam penelitian ini load balancer web menggunakan Nginx sedangkan load balancer database menggunakan HaProxy dan MaxScale.



Gambar 12. Arsitektur Baru Moodle

4. Membuat Prototype Arsitektur

Hanya ada dua komponen yang terhubung langsung dengan internet yaitu web load balancer dan gateway. Web load balancer menjadi gerbang pertama masuknya request dari pengguna. Request yang masuk kemudian diteruskan kepada web server. Koneksi yang digunakan dari load balancer ke web server dan dari web server menuju database server merupakan koneksi private menggunakan IP private.

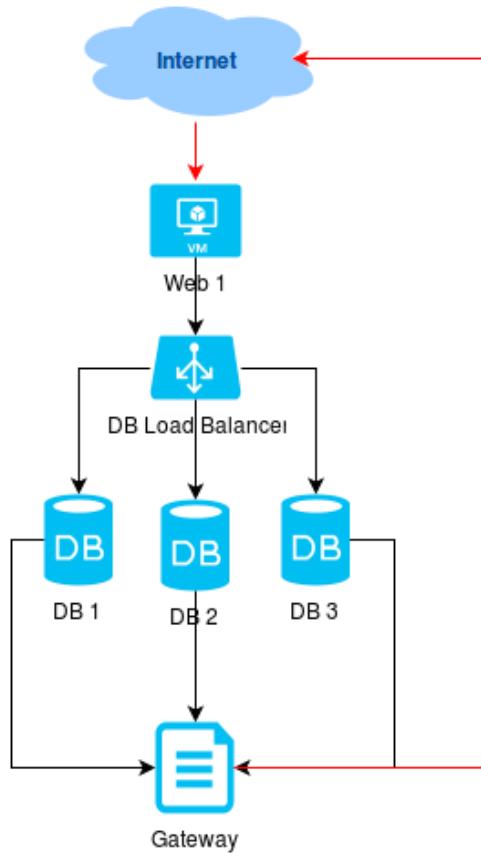


Gambar 13. Arsitektur Baru Jaringan Moodle

5. Revisi Prototype Arsitektur Baru

Setelah dilakukan ujicoba, didapatkan data ternyata untuk ujicoba load balancer tidak membutuhkan banyak web server karena load web server yang

masih rendah sehingga akhirnya web server dikurangi dari 3 web server menjadi 1 web server.



Gambar 14. Arsitektur Moodle dalam Penelitian

C. Membangun Infrastruktur Menggunakan Arsitektur Baru

Ada beberapa komponen yang harus dikonfigurasi saat membangun arsitektur baru. Arsitektur baru berjalan di datacenter AWS yang menggunakan teknologi cloud. Server berjalan sebagai virtual machine bukan sebagai server fisik. Menggunakan virtual machine mempunyai keuntungan namun juga mempunyai kerugian, keuntungannya peneliti tidak harus memanajemen server fisik yang rawan mengalami kerusakan, sedangkan jika menggunakan teknologi cloud dari

AWS, peneliti hanya perlu membuat VM tanpa harus memanajemen server fisik.

Saat server fisik yang digunakan AWS bermasalah, teknologi cloud dari AWS memungkinkan VM tetap hidup menggunakan server fisik lainnya. Perpindahan dari server fisik ke server fisik lainnya terjadi secara otomatis.

1. Menggunakan layanan EC2

Penelitian ini hanya menggunakan layanan EC2 dan tidak menggunakan layanan yang lainnya.

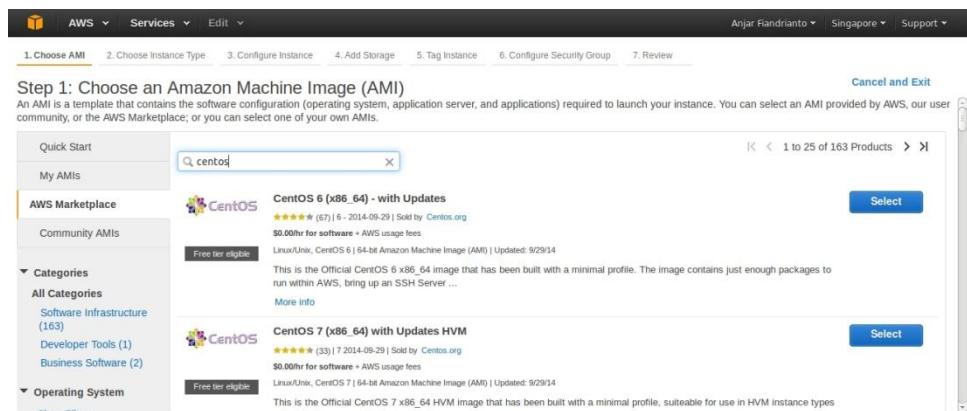


Gambar 15. Halaman Layanan AWS

Layanan EC2 mempunyai banyak region (data center) yang bisa digunakan, pemilihan region disesuaikan dengan kebutuhan. Memilih region berarti memilih dimana datacenter yang dipilih. Dalam penelitian ini, peneliti menggunakan datacenter Singapura untuk menempatkan VM. Datacenter Singapura dipilih karena lokasinya yang dekat dengan Indonesia sehingga *latency* rendah. *Latency* rendah membuat VM lebih cepat diakses, semakin cepat VM diakses, semakin cepat proses penggerjaan penelitian ini.

2. Sistem Operasi

Sistem operasi yang tersedia di AWS terbagi menjadi dua yaitu yang berbayar dan gratis. Sistem operasi berbayar seperti Windows Server dan RedHat. Sementara CentOS gratis karena berbasis komunitas.



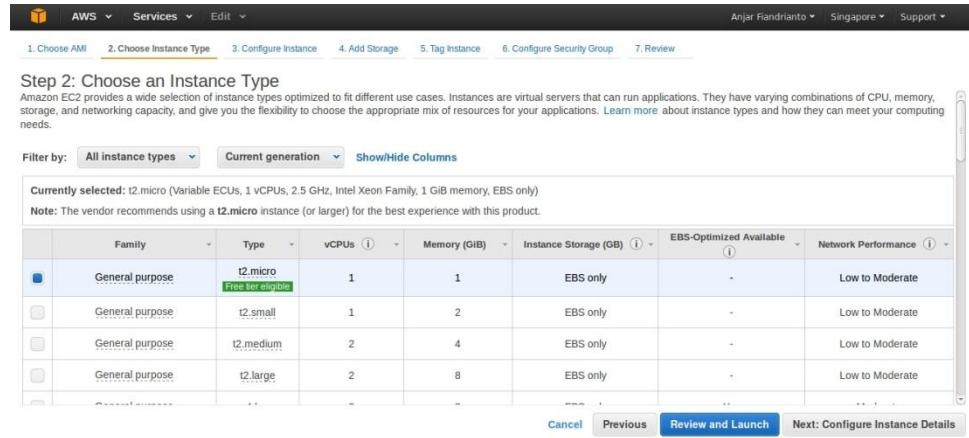
Gambar 16. Pilihan Sistem Operasi AWS

Penelitian ini menggunakan sistem operasi CentOS 6 versi terakhir.

3. Spesifikasi Virtual Machine

Jenis vm terbagi menjadi banyak jenis tergantung dari alokasi RAM, CPU, kualitas jaringan, dan kualitas IO ke media penyimpanan. Peneliti bisa memilih berapa RAM, CPU dan aspek lain yang dibutuhkan setiap VM untuk bisa menyesuaikan kebutuhan dan biaya.

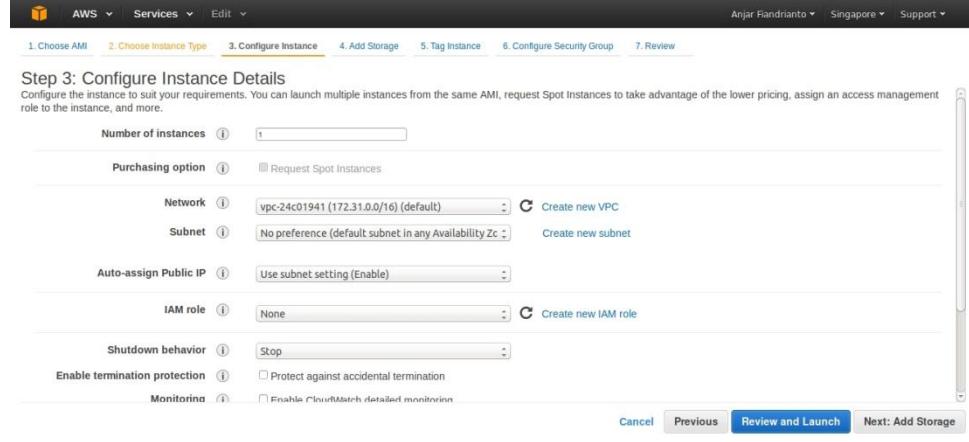
Dalam penelitian ini peneliti menggunakan dua jenis VM yaitu t2.micro dan t2.large. VM t2.micro digunakan untuk load balancer database server dan server benchmark sedangkan t2.large digunakan untuk web server.



Gambar 17. Jenis-Jenis Virtual Machine di AWS

4. Spesifikasi jaringan

Dalam penelitian ini, spesifikasi perangkat network mengikuti spesifikasi standar dari Amazon. Jaringan menggunakan VLAN dengan subnet 172.31.0.0/16 dengan IP Private diperoleh secara DHCP.



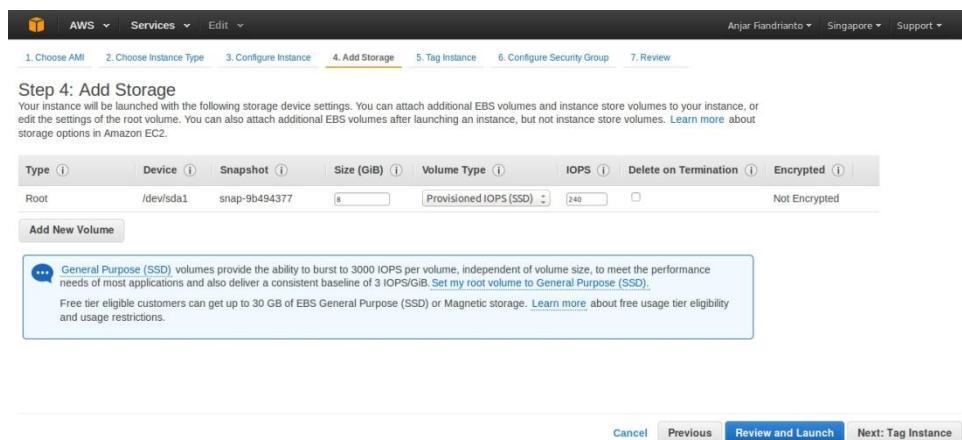
Gambar 18. Konfigurasi Jaringan

5. Spesifikasi Storage

Storage di AWS terbagi menjadi tiga yaitu Magnetic, general purpose (SSD) dan provisioned IOPS (SSD). Tipe storage magnetic menggunakan hard disk sebagai

media penyimpanan sedangkan storage general purpose menggunakan SSD sebagai media penyimpanan. Provisioned IOPS berbeda dengan jenis general purpose walaupun sama-sama menggunakan SSD karena provisioned IOPS memberikan garansi IOPS, tidak seperti jenis storage lain dimana menggunakan *shared* IOPS. Provisioned IOPS memberikan garansi yang artinya performa IO VM tersebut tidak akan dibawah nilai IOPS yang sudah dipesan.

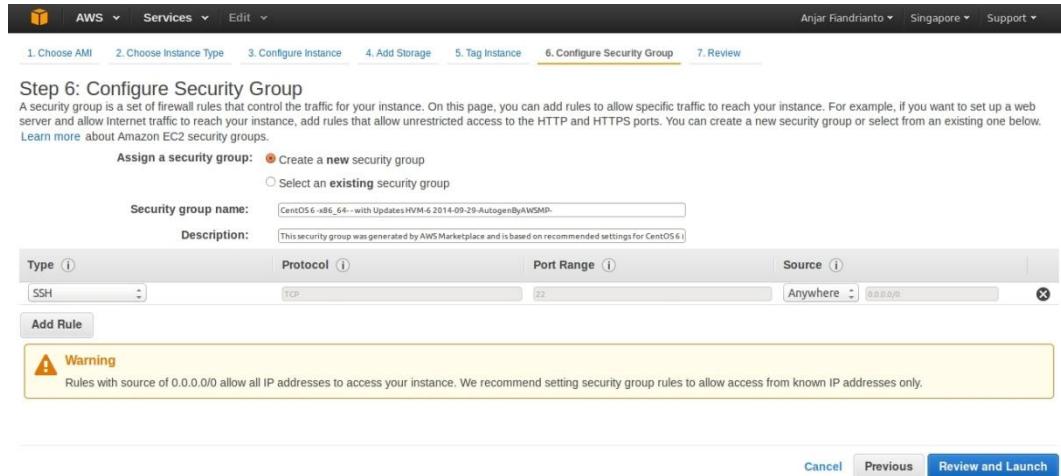
Penelitian ini menggunakan storege dengan jenis General Purpose (SSD) karena secara kualitas masih sangat mencukupi untuk kebutuhan benchmark. Peneliti memilih kapasitas storage minimal yaitu 8GB karena hanya akan diisi sistem operasi dan Moodle yang penggunaan totalnya masih dibawah 3GB.



Gambar 19. Spesifikasi Storage VM AWS

6. Konfigurasi keamanan Virtual Machine

Jaringan di AWS menggunakan NAT, IP public yang diarahkan ke server tidak dikonfigurasi langsung di VM tapi harus dikonfigurasi di management console. Jika menggunakan AWS, konfigurasi *IP Tables* tidak hanya di sistem operasi namun juga di router AWS.



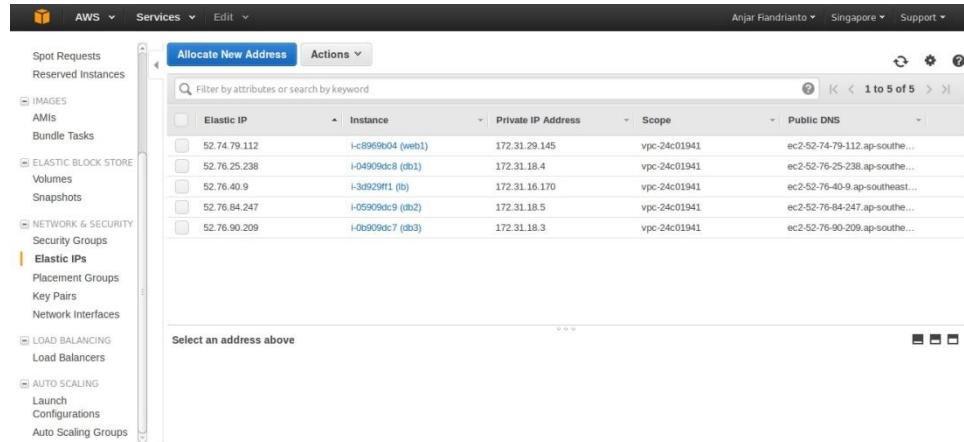
Gambar 20. Konfigurasi Firewall VM AWS

Peneliti menentukan port apa saja yang harus dibuka dan ditutup aksesnya dari luar. Port yang dibuka dalam penelitian ini adalah port SSH dan port 80 sebagai akses web server.

7. Konfigurasi IP Publik

Secara *default* semua VM yang ada di amazon tidak mempunyai IP public statik. AWS menyediakan layanan IP publik yaitu elastic IP. Pengguna AWS bisa menyewa IP untuk dialokasikan untuk setiap VM yang ada di cloud. Setiap VM di AWS saat selesai install hanya mempunyai IP private sedangkan untuk koneksi dari luar dibutuhkan IP publik. Penelitian ini membutuhkan 5 IP publik yang dialokasikan untuk setiap VM. Berikut daftar IP publik yang dialokasikan ke setiap VM

Dalam gambar di atas, terlihat ada 5 IP public yang disewa dan sudah dialokasikan ke setiap VM yang digunakan untuk membuat cluster moodle. Alasan fleksibilitas pengguna IP ini juga membuat peneliti memilih AWS sebagai penyedia layanan cloud.



The screenshot shows the AWS Management Console with the 'AWS Services' bar at the top. The 'Elastic IPs' section is selected in the left sidebar. A table lists five assigned IP addresses, each with a checkbox, instance ID, private IP address, scope, and public DNS. The table has columns for 'Elastic IP', 'Instance', 'Private IP Address', 'Scope', and 'Public DNS'. A search bar at the top of the table says 'Filter by attributes or search by keyword'. Below the table, a message says 'Select an address above' with three small icons.

Elastic IP	Instance	Private IP Address	Scope	Public DNS
52.74.79.112	i-c8969b04 (web1)	172.31.29.145	vpc-24c01941	ec2-52-74-79-112.ap-southe...
52.76.25.238	i-04909de8 (db1)	172.31.18.4	vpc-24c01941	ec2-52-76-25-238.ap-southe...
52.76.40.9	i-3d929ff1 (lb)	172.31.16.170	vpc-24c01941	ec2-52-76-40-9.ap-southeast...
52.76.64.247	i-05909de9 (db2)	172.31.18.5	vpc-24c01941	ec2-52-76-64-247.ap-southe...
52.76.90.209	i-0b909dc7 (db3)	172.31.18.3	vpc-24c01941	ec2-52-76-90-209.ap-southe...

Gambar 21. Detail Alamat IP VM AWS

8. Konfigurasi Global Server

Setiap server harus dikonfigurasi manual agar bisa memenuhi kebutuhan spesifik tiap VM. Sebelum diinstall menggunakan software spesifik, terlebih dahulu server dikonfigurasi secara global. Berikut beberapa konfigurasi yang dilakukan,

- Mengubah SELINUX dari *enforcing* ke *disabled*
- Konfigurasi Repo
- Konfigurasi NTP
- Konfigurasi Software Monitoring

9. Database Server (Percona XtraDB Cluster)

Database server menggunakan Percona XtraDB Cluster. Database server dalam penelitian ini berbeda dari MySQL biasa karena database dibangun dalam model cluster dan master-to-master sehingga semua database server bisa digunakan untuk melulis data (active-active).

10. Web Server (Apache Web Server dan PHP)

Web server yang digunakan dalam penelitian ini menggunakan Apache 2.2.15. Menggunakan Apache karena kompatibilitas moodle paling baik jika web server menggunakan Apache. Selain Apache, moodle juga membutuhkan PHP 5. PHP berfungsi untuk menterjemahkan *source code* PHP ke kode binary yang bisa langsung dieksekusi oleh mesin.

11. Load Balancer (HaProxy dan MaxScale)

Ada dua load balancer yang digunakan dalam penelitian ini. HaProxy merupakan TCP load balancer yang sebenarnya tidak hanya untuk database load balancer namun juga bisa digunakan sebagai load balancer aplikasi lain yang berbasis koneksi TCP.

D. Hasil Pengujian Load Balancer

Pengujian load balancer dilakukan dengan benchmark menggunakan software httpperf. Berikut perintah yang digunakan untuk melakukan benchmark

```
httpperf --server skrip.anjar.web.id --uri /moodle/mod/page/view.php?id=4 --num-conn 384 --timeout 10 --rate 10
```

Ada beberapa varibel yang harus disesuaikan dengan instrumen penelitian, variabel-variabelnya sebagai berikut:

1. --server

Variabel --server merupakan variabel yang harus diisi alamat server tanpa path web. Dalam penelitian ini server yang digunakan menggunakan domain skrip.anjar.web.id.

2. --uri

Variabel --uri merupakan variabel yang berisi path URL. Dalam penelitian ini path yang digunakan merupakan halaman mata pelajaran di moodle. Menggunakan halaman mata pelajaran karena jika diakses halaman ini mempunyai kombinasi query write dan read.

3. --num-conn

Variabel --um-con merupakan variabel yang diisi dengan jumlah sampel. Penelitian ini menggunakan sampel sejumlah 384 sampel.

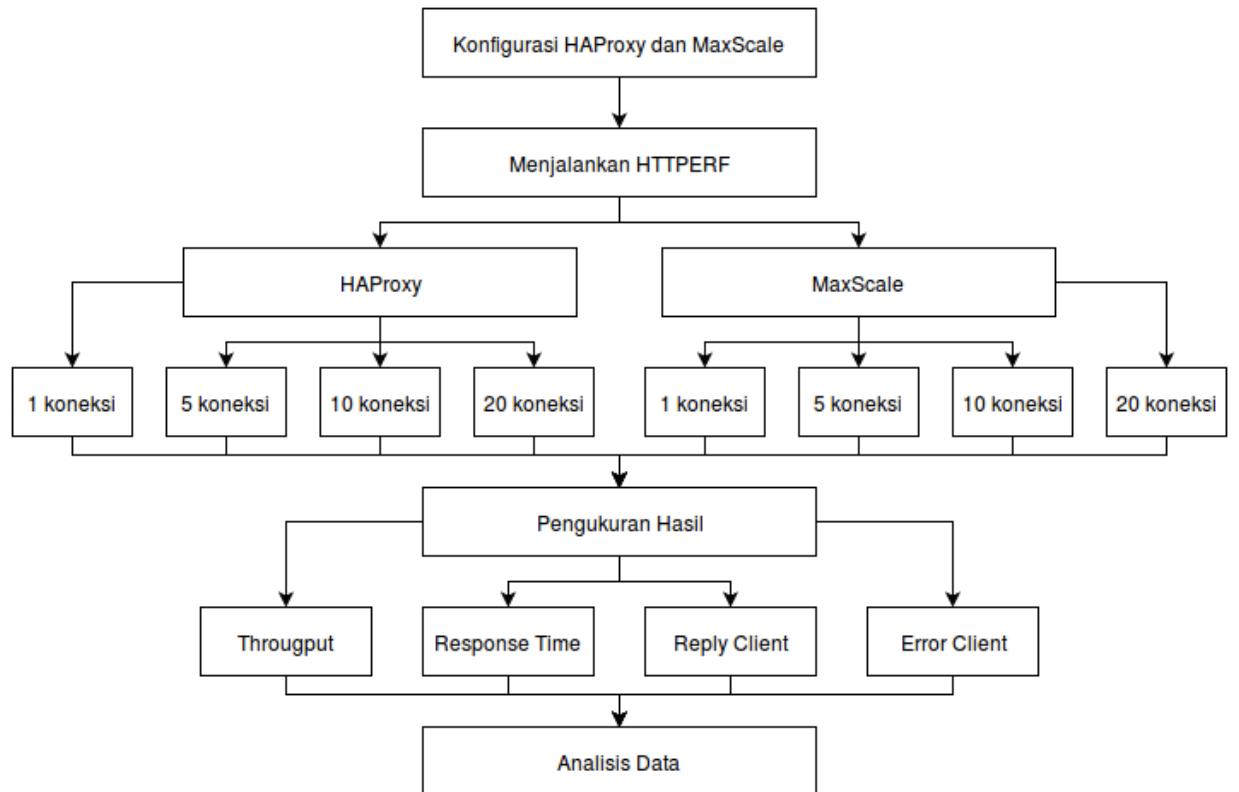
4. --timeout

Variabel timeout adalah batas waktu loading yang dibutuhkan oleh web server untuk menyelesaikan proses render halaman web yang dikirimkan oleh software benchmark. Dalam penelitian ini timeout dikonfigurasi menjadi 10 detik. Didapat angka 10 detik karena disesuaikan dengan standar ITU-T G.1030 11/2005 tentang waktu *loss of attention*. Dalam ITU-T G.1030 11/2005 disebutkan bahwa 10 detik adalah batasan waktu pengguna website tetap fokus menunggu hasil render dari web server.

5. --rate

Variabel --rate adalah variable yang digunakan untuk membuat paralel proses dalam setiap detik. Dalam penelitian ini menggunakan beberapa nilai yaitu 1, 5, 10, 15, 20, 25 dan 30.

Berikut bagan proses dalam melakukan pengujian load balancer:

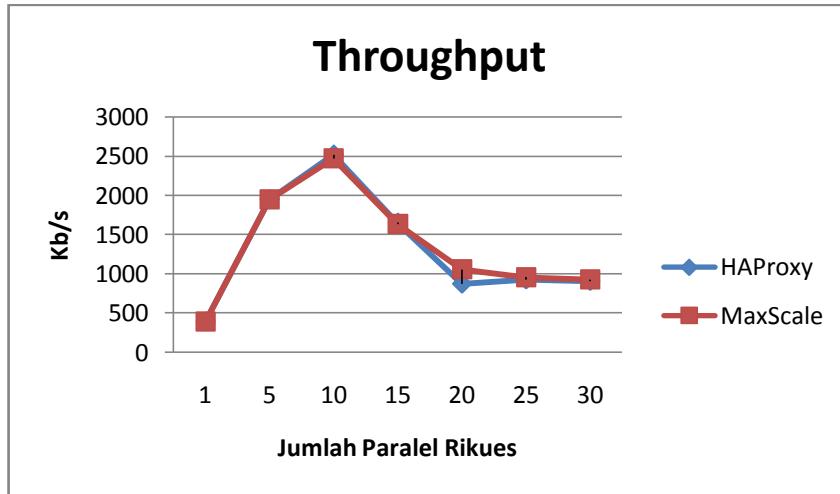


Gambar 22. Proses Pengujian Load Balancer

1. Hasil Pengujian Throughput

Jumlah Paralel Proses	HAProxy (kbps)	MaxScale (kbps)
1	392.8	392.8
5	1948.8	1947.8
10	2518.9	2470.7
15	1650.9	1634.5
20	874.2	1057.6
25	928	957.6
30	909.2	926.4

Tabel 1. Hasil Pengujian Throughput



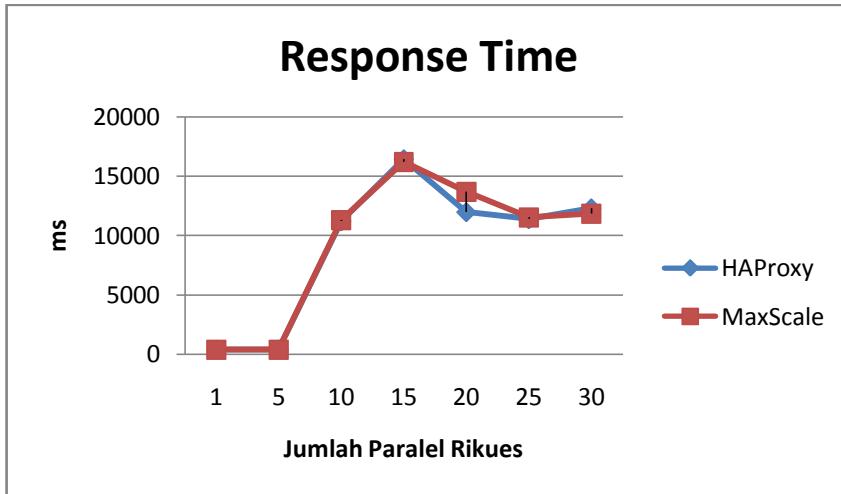
Gambar 23. Hasil Pengujian throughput

Throughput adalah jumlah data (kb) yang lewat dalam satu satuan waktu (per detik). Throughput menggunakan satuan kbps, yaitu berapa kilobyte data yang lewat dalam satu detik. Throughput satu request moodle tanpa koneksi lain adalah kurang lebih 400 kbps, namun bertambah saat jumlah paralel koneksi ditambah. Saat request 1 koneksi setiap detik sampai 10 request setiap detik, sistem masih berjalan secara normal, namun saat jumlah paralel request sudah mencapai 15 request setiap detik, kondisi sistem mulai tidak stabil.

2. Hasil Pengujian Response Time

Jumlah Paralel Proses	HAProxy (ms)	MaxScale (ms)
1	364.3	373.5
5	376.2	390.7
10	11203.2	11319.2
15	16435.6	16200.2
20	11977.5	13701.2
25	11388.6	11530.7
30	12295.9	11836.9

Tabel 2. Hasil Pengujian Response time



Gambar 24. Hasil Pengujian Response Time

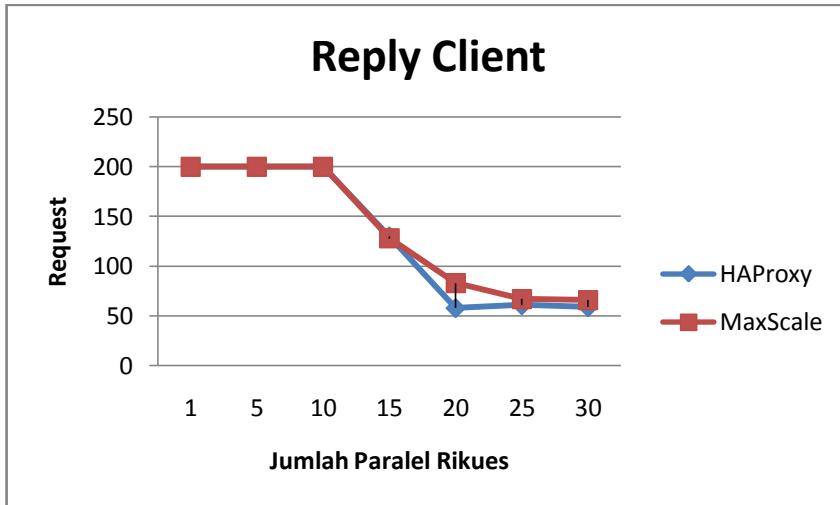
Response time diukur menggunakan satuan mili detik (milisecond).

Response time merupakan waktu yang dibutuhkan pengguna untuk menunggu untuk mendapatkan halaman yang diakses. Dalam penelitian ini, response time moodle merupakan waktu yang dibutuhkan server cluster untuk menerima dan mengembalikan halaman sesuai yang diakses pengguna. Response time adalah selang waktu antara user memasukkan suatu perintah ke dalam sistem hingga mendapat balasan selengkapnya dari sistem. Saat request mencapai 10 request setiap detik, sistem rata-rata server mengembalikan data dalam waktu 11 detik.

3. Hasil Pengujian Reply Client

Jumlah Paralel Proses	HAProxy (request)	MaxScale (request)
1	200	200
5	200	200
10	200	200
15	130	128
20	58	83
25	61	67
30	59	66

Tabel 3. Hasil Pengujian Reply Client



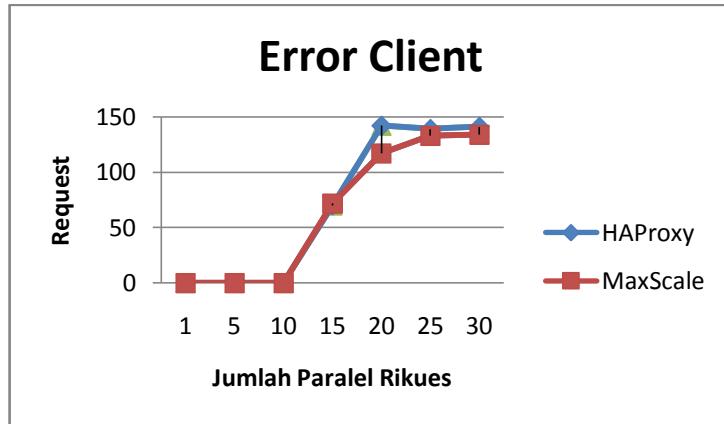
Gambar 25. Hasil Pengujian Reply Client

Reply client adalah jumlah halaman yang berhasil diterima oleh penggunaan (software benchmark) dalam kondisi baik. Halaman yang diterima dalam kondisi baik mempunyai tanda menggunakan HTTP status dengan nilai 200. Timeout setetapkan sebesar 10 detik, setelah 10 detik tidak selesai, maka dianggap gagal mengembalikan request dan dihitung sebagai request error. Sistem masih konsisten saat dibebani request antara 1-10 request setiap detik namun mulai tidak stabil saat request 15 sampai 30 request setiap detik.

4. Hasil Pengujian Error Client

Jumlah Paralel Proses	HAProxy (request)	MaxScale (request)
1	0	0
5	0	0
10	0	0
15	70	72
20	142	117
25	139	133
30	141	134

Tabel 4. Hasil Pengujian Error Client



Gambar 26. Hasil Pengujian Error Client

Error client adalah jumlah halaman yang gagal diterima oleh pengguna (software benchmark) dalam kondisi baik. Halaman yang diterima dalam kondisi gagal mempunyai tanda HTTP status dengan nilai 400 dan 500. Timeout setetapkan sebesar 10 detik, setelah 10 detik tidak selesai, maka dianggap gagal mengembalikan request dan dihitung sebagai request error. Error client dalam observasi menunjukkan bahwa tingkat kegagalan pengembalian request lebih rendah untuk MaxScale, hasil ini setara dengan hasil dalam ujicoba Reply Client.

E. Analisis Data Hasil Pengujian

Analisis penelitian ini menggunakan perangkat lunak Microsoft Excel dengan *add-on* Analysis Toolpack. Berikut hasil analisis data yang diperoleh dari pengujian sebelumnya,

1. Analysis Throughput

	A	B	C
1	t-Test: Paired Two Sample for Means		
2			
3		HAProxy	MaxScale
4	Mean	1317.543	1341.057
5	Variance	553843.6	504809.7
6	Observations	7	7
7	Pearson Correlation	0.99579	
8	Hypothesized Mean Difference	0	
9	df	6	
10	t Stat	-0.83222	
11	P(T<=t) one-tail	0.218574	
12	t Critical one-tail	1.94318	
13	P(T<=t) two-tail	0.437148	
14	t Critical two-tail	2.446912	

Gambar 27. Hasil Analisis Throughput

Pada gambar diatas, terlihat bahwa rata-rata throughput dari HAProxy dan MaxScale ada perbedaan namun tidak signifikan. Nilai $P(T \leq t)$ two-tail lebih besar dari taraf nyata 5% yaitu sebesar 0.437148 sehingga bisa disimpulkan bahwa tidak terdapat perbedaan yang signifikan antara HAProxy dan MaxScale.

2. Analysis Response Time

	A	B	C
1	t-Test: Paired Two Sample for Means		
2			
3		HAProxy	MaxScale
4	Mean	9148.757	9336.057
5	Variance	39061075	40253550
6	Observations	7	7
7	Pearson Correlation	0.993754	
8	Hypothesized Mean Difference	0	
9	df	6	
10	t Stat	-0.69784	
11	P(T<=t) one-tail	0.2557	
12	t Critical one-tail	1.94318	
13	P(T<=t) two-tail	0.511401	
14	t Critical two-tail	2.446912	

Gambar 28. Hasil Analisis Response Time

Pada gambar diatas, terlihat bahwa rata-rata response time dari HAProxy dan MaxScale ada perbedaan namun tidak signifikan. Nilai $P(T \leq t)$ two-tail lebih besar dari taraf nyata 5% yaitu sebesar 0.511401 sehingga bisa disimpulkan bahwa tidak terdapat perbedaan yang signifikan dalam *response time* antara HAProxy dan MaxScale.

3. Analysis Reply Client

	A	B	C
1	t-Test: Paired Two Sample for Means		
2			
3		HAProxy	MaxScale
4	Mean	257.1429	256.7143
5	Variance	15992.81	15893.9
6	Observations	7	7
7	Pearson Correlation	0.997936	
8	Hypothesized Mean Difference	0	
9	df	6	
10	t Stat	0.139623	
11	P(T<=t) one-tail	0.446763	
12	t Critical one-tail	1.94318	
13	P(T<=t) two-tail	0.893527	
14	t Critical two-tail	2.446912	

Gambar 29. Hasil Analisis Reply Client

Pada gambar diatas, terlihat bahwa rata-rata reply client dari HAProxy dan MaxScale ada perbedaan namun tidak signifikan. Nilai $P(T \leq t)$ two-tail lebih besar dari taraf nyata 5% yaitu sebesar 0.893527 sehingga bisa disimpulkan bahwa tidak terdapat perbedaan yang signifikan dalam reply client antara HAProxy dan MaxScale.

4. Analysis Error Client

Pada gambar dibawah, terlihat bahwa rata-rata error client dari HAProxy dan MaxScale ada perbedaan namun tidak signifikan. Nilai $P(T \leq t)$ two-tail lebih besar dari taraf nyata 5% yaitu sebesar 0.928115 sehingga bisa disimpulkan bahwa tidak terdapat perbedaan yang signifikan dalam error client antara HAProxy dan MaxScale.

	A	B	C
1	t-Test: Paired Two Sample for Means		
2			
3		HAProxy	MaxScale
4	Mean	127	127.2857
5	Variance	15994	15893.9
6	Observations	7	7
7	Pearson Correlation	0.99798	
8	Hypothesized Mean Difference	0	
9	df	6	
10	t Stat	-0.09407	
11	P(T<=t) one-tail	0.464057	
12	t Critical one-tail	1.94318	
13	P(T<=t) two-tail	0.928115	
14	t Critical two-tail	2.446912	

Gambar 30. Hasil Analisis Error Client

F. Pembahasan Hasil Penelitian

1. Throughput

Berdasarkan hasil ujicoba, throughput satu request moodle tanpa koneksi lain adalah kurang lebih 400kbps, namun bertambah saat jumlah paralel koneksi ditambah. Saat request 1 koneksi setiap detik sampai 10 request setiap detik, sistem masih berjalan secara normal, namun saat jumlah paralel request sudah mencapai 15 request setiap detik, kondisi sistem mulai tidak stabil. Saat sistem masih normal, HAProxy menghasilkan jumlah throughput yang lebih tinggi namun saat jumlah paralel koneksi sudah banyak, MaxScale menunjukkan throughput yang lebih baik. Kondisi ini disebabkan oleh HAProxy yang belum mendukung multi-core. MaxScale mempunyai fitur multithreaded, yaitu fitur yang memungkinkan MaxScale menggunakan semua core CPU yang ada di server.

2. Response time

Saat request masih 1 sampai 5 request setiap detik nilai response time masih sekitar 200ms menunjukkan sistem masih dalam kondisi normal. Saat request mencapai 10 request setiap detik, sistem rata-rata mengembalikan data dalam waktu 11 detik. Walaupun sudah 10 detik, namun sistem masih berjalan normal karena dari nilai reply client, semua request yang dikirim oleh server benchmark masih dikembalikan secara penuh. Namun saat request sudah mencapai 15 request setiap detik, mulai terjadi bottleneck yang menyebabkan sistem menjadi tidak stabil. Kondisi paling parah adalah saat sistem diberi beban 20-30 request setiap detik, terlihat terjadi bottleneck yang parah karena server tidak bisa mengembalikan semua request yang dikirim oleh server benchmark.

3. Reply Client

Dalam penelitian ini, timeout setetapkan sebesar 10 detik, setelah 10 detik tidak selesai, maka dianggap gagal mengembalikan request. Sistem masih konsisten saat dibebani request antara 1-10 request setiap detik namun mulai tidak stabil saat request 15 sampai 30 request setiap detik. Saat request 15 sampai 30, MaxScale memberikan hasil yang lebih baik daripada HAProxy karena tingkat keberhasilan mengembalikan request yang lebih baik. Kondisi server yang overload memaksa kinerja HAProxy yang hanya menggunakan 1 core CPU kurang optimal.

4. Error Client

Error client dalam observasi menunjukkan bahwa tingkat kegagalan pengembalian request lebih rendah untuk MaxScale, hasil ini setara dengan hasil dalam ujicoba Reply Client. Kondisi ini disebabkan karena HAProxy yang tidak

mendukung multithreaded. Jika dilihat dari perangkat lunak monitoring, terlihat bahwa HAProxy hanya menggunakan satu core processor untuk pemrosesan request.

BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian dan pembahasan maka peneliti dapat mengambil kesimpulan sebagai berikut:

1. Moodle bisa dijadikan sistem cluster dengan menjadikan web server tunggal menjadi cluster dan database server tunggal menjadi database cluster. Database cluster menggunakan Galera Cluster dan Percona XtraDB Cluster. Dengan menggunakan database cluster, skalabilitas database server menjadi lebih baik.
2. Load balancer MaxScale mempunyai fitur read-write split yang menjadikan database server tidak mengalami deadlock sehingga database terbebas dari masalah redundansi data.
3. Tidak ditemukan perbedaan performa yang signifikan untuk website Moodle jika menggunakan load balancer HAProxy atau MaxScale.
Performa HaProxy lebih baik saat kondisi server dalam beban yang rendah namun MaxScale lebih stabil saat server dalam kondisi beban yang berat. Namun secara keseluruhan tidak ada perbedaan tidak signifikan.

B. Saran

Berdasarkan dari simpulan dan temuan dari penelitian yang telah dilakukan, maka peneliti memberikan saran sebagai berikut:

1. Untuk mencapai tingkat skalabilitas yang tinggi, Moodle dari awal sebaiknya dikonfigurasi menggunakan sistem cluster.

2. Untuk menghindari masalah redudansi data di Moodle, disarankan menggunakan load balancer MaxScale. Selain tidak terjadi redudansi data, MaxScale juga lebih stabil saat beban server tinggi.
3. Dengan fitur MaxScale yang lebih baik, MaxScale disarankan digunakan untuk menggantikan HAProxy sebagai load balancer.

DAFTAR PUSTAKA

- Aribowo, Didik. 2013. Optimalisasi Cluster Server LMS dan IPTV dengan Variasi algoritma Penjadwalan
- Badger dkk. 2012. Cloud Computing Synopsis and Recommendations
- Bourke, Tony. 2001. Server Load Balancing
- Chandrasekaran, K. 2015. Essentials of Cloud Computing
- Jayaswal, Kailash. 2006. Administering Data Centers: Servers, Storage, and Voice over IP
- Jin, Tai. 1998. Htpperf: A Tool for Measuring Web Server Performance
- ITU-T G.1030 11/2005. Estimating end-to-end performance in IP networks for data applications.
- Kooper. 2005. Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software
- Muhammad, Omar Altoumi Alsyabani.2013. Performa Algoritma Load Balance Pada Server Web Apache dan Nginx dengan Database Postgresql
- Morrison, Richard S. 2003. Cluster Computing: Architectures, Operating Systems, Parallel Processing & Programming Languages. GNU General Public Licence.
- Nazir, Moh. 2005. Metode penelitian
- Orsini, Rob. 2007. Rails Cookbook. Sebastopol. O'Reilly Media, Inc
- Paul, E. Black. 2005. Dictionary of Algorithms and Data Structures
- Pfister, Gregory. 2007. In Search of Clusters
- Pierre, Mavro. 2014. MariaDB High Performance
- Shroff, Gautam. 2010. Enterprise Cloud Computing : Technology, Architecture, Application
- Sirajuddin dkk. 2012. Rancang Bangun Server Learning Management System Menggunakan Load Balancer dan Reverse Proxy

Sugiyono. (2013). Statistika untuk Penelitian. Bandung: Alfabeta.

Tafiardi. 2005. Meningkatkan mutu pendidikan melalui e-learning

Xiaojie, Yang . 2011. Analysis of DBMS: MySQL Vs PostgreSQL

LAMPIRAN

Lampiran 1. File Konfigurasi Server Database Server 1

```
[mysqld]
datadir=/var/lib/mysql
user=mysql
wsrep_provider=/usr/lib64/libgalera_smm.so
wsrep_cluster_address=gcomm://172.31.18.4,172.31.18.5,172.31.18.3
#wsrep_cluster_address=gcomm://
binlog_format=ROW
default_storage_engine=InnoDB
innodb_locks_unsafe_for_binlog=1
innodb_autoinc_lock_mode=2
wsrep_node_address=172.31.18.4
wsrep_sst_method=xtrabackup-v2
wsrep_cluster_name=my_centos_cluster
wsrep_sst_auth="usernyagalerarep:passwordgalera*"
```

Lampiran 2. File Konfigurasi Server Database Server 2

```
[mysqld]
datadir=/var/lib/mysql
user=mysql
wsrep_provider=/usr/lib64/libgalera_smm.so
wsrep_cluster_address=gcomm://172.31.18.4,172.31.18.5,172.31.18.3
#wsrep_cluster_address=gcomm://
#wsrep_cluster_address=gcomm://172.31.18.4
binlog_format=ROW
default_storage_engine=InnoDB
innodb_locks_unsafe_for_binlog=1
innodb_autoinc_lock_mode=2
wsrep_node_address=172.31.18.5
wsrep_sst_method=xtrabackup-v2
wsrep_cluster_name=my_centos_cluster
wsrep_sst_auth="usernyagalerarep:passwordgalera*"
```

Lampiran 3. File Konfigurasi Server Database Server 3

```
[mysqld]
datadir=/var/lib/mysql
user=mysql
wsrep_provider=/usr/lib64/libgalera_smm.so
wsrep_cluster_address=gcomm://172.31.18.4,172.31.18.5,172.31.18.3
#wsrep_cluster_address=gcomm://
#wsrep_cluster_address=gcomm://172.31.18.4
binlog_format=ROW
default_storage_engine=InnoDB
innodb_locks_unsafe_for_binlog=1
innodb_autoinc_lock_mode=2
wsrep_node_address=172.31.18.3
wsrep_sst_method=xtrabackup-v2
wsrep_cluster_name=my_centos_cluster
wsrep_sst_auth="usernyagalerarep:passwordgalera*"
```

Lampiran 4. File Konfigurasi Server HAProxy

HaProxy

```
global
```

```
    log 127.0.0.1    local0
    log 127.0.0.1    local1 notice
    maxconn 40000
    user haproxy
    group haproxy
    daemon
```

```
defaults
```

```
    log    global
    mode   http
    retries 3
    option  redispatch
    maxconn 400
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms
```

```
listen DB 0.0.0.0:21221
```

```
    mode tcp
    balance roundrobin
    maxconn 3500
    server db1 172.31.18.4:3306 maxconn 1000 check
    server db2 172.31.18.5:3306 maxconn 1000 check
```

```
server db3 172.31.18.3:3306 maxconn 1000 check
```

```
listen admin
  bind 0.0.0.0:8080
  stats enable
  stats enable
  stats hide-version
  stats uri  /stats
  stats auth  anjarwebid:1234
```

Lampiran 5. File Konfigurasi Server MaxScale

```
[maxscale]
threads=4

[Splitter Service]
type=service
router=readwritesplit
servers=dbserv1, dbserv2, dbserv3
user=root
passwd=syan
enable_root_user=1

[Splitter Listener]
type=listener
service=Splitter Service
protocol=MySQLClient
port=3306
#address=0.0.0.0
#socket=/tmp/socket

[dbserv1]
type=server
address=172.31.18.4
port=3306
protocol=MySQLBackend

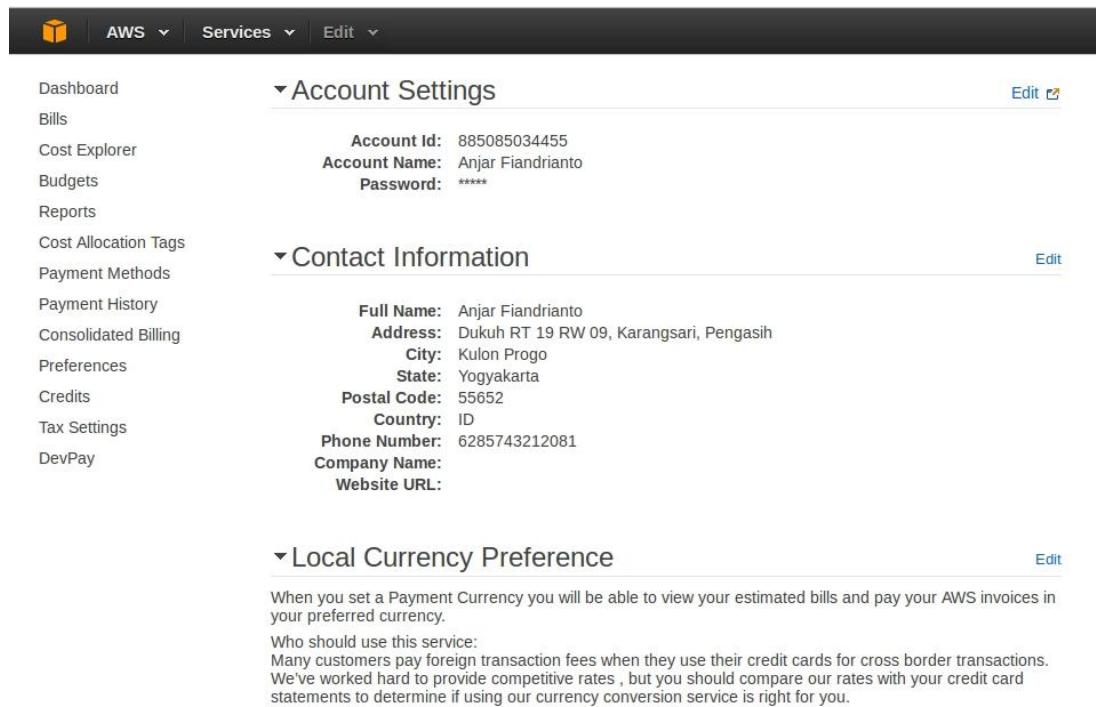
[dbserv2]
type=server
address=172.31.18.5
port=3306
```

```
protocol=MySQLBackend
[dbserv3]
type=server
address=172.31.18.3
port=3306
protocol=MySQLBackend
[Galera Monitor]
type=monitor
module=galeramon
servers=dbserv1, dbserv2, dbserv3
user=root
passwd=syan
[CLI]
type=service
router=cli
[CLI Listener]
type=listener
service=CLI
protocol=maxscaled
address=localhost
port=6603
```

Lampiran 6. File Konfigurasi Moodle

```
$CFG->dbtype    = 'mysqli';
$CFG->dblibrary = 'native';
$CFG->dbhost    = '172.31.16.170';
$CFG->dbname    = 'moodle';
$CFG->dbuser    = 'root';
$CFG->dbpass    = 'syan';
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
    'dbpersist' => 0,
    'dbport'    => 21221,
    'dbsocket'  => '',
);
$CFG->wwwroot   = 'http://skrip.anjar.web.id/moodle';
$CFG->dataroot  = '/var/www/moodledata';
$CFG->admin     = 'admin';
$CFG->directorypermissions = 0777;
```

Lampiran 7. Halaman Akun Amazon Web Services



The screenshot shows the AWS Account Settings page. The top navigation bar includes the AWS logo, a dropdown for 'AWS', a 'Services' dropdown, and an 'Edit' dropdown. The left sidebar lists various account management options: Dashboard, Bills, Cost Explorer, Budgets, Reports, Cost Allocation Tags, Payment Methods, Payment History, Consolidated Billing, Preferences, Credits, Tax Settings, and DevPay. The main content area is divided into sections: 'Account Settings' (Account Id: 885085034455, Account Name: Anjar Fiandrianto, Password: ****, with an 'Edit' link), 'Contact Information' (Full Name: Anjar Fiandrianto, Address: Dukuh RT 19 RW 09, Karangsari, Pengasih, City: Kulon Progo, State: Yogyakarta, Postal Code: 55652, Country: ID, Phone Number: 6285743212081, Company Name: [redacted], Website URL: [redacted], with an 'Edit' link), and 'Local Currency Preference' (a brief description about payment currency, with an 'Edit' link). The 'Edit' links in each section are accompanied by a small edit icon.