TESIS – TE142599

IMPLEMENTASI DAN EVALUASI KINERJA DISASTER RECOVERY DALAM LINGKUNGAN DATA CENTER BERBASIS CLOUD

YORISAN PERMANA BAGINDA
07111350030014

DOSEN PEMBIMBING
Dr. Istas Pratomo S.T., M.T.
Dr. Ir. Achmad Affandi, DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

TESIS – TE142599

IMPLEMENTASI DAN EVALUASI KINERJA DISASTER RECOVERY DALAM LINGKUNGAN DATA CENTER BERBASIS CLOUD

YORISAN PERMANA BAGINDA
07111350030014

DOSEN PEMBIMBING
Dr. Istas Pratomo S.T., M.T.
Dr. Ir. Achmad Affandi, DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Yorisan Permana Baginda
NRP. 07111350030014

Tanggal Ujian : 5 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:

1. Dr. Istas Pratomo S.T., M.T (Pembimbing I)
   NIP: 197903252003121001
2. Dr. Ir. Achmad Affandi, DEA (Pembimbing II)
   NIP: 196510141990021001
3. Dr. Ir. Endroyono, DEA. (Penguji)
   NIP: 196504041991021001
4. Eko Setijadi, ST., MT., Ph.D. (Penguji)
   NIP: 197210012003121002
5. Dr. Ir. Achmad Mauludiyanto, MT. (Penguji)
   NIP: 196109031989031001

Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**IMPLEMENTASI DAN EVALUASI KINERJA DISASTER RECOVERY DALAM LINGKUNGAN DATA CENTER BERBASIS CLOUD**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Maret 2018

Yorisan Permana Baginda
NRP. 07111350030014

Halaman ini sengaja dikosongkan

IMPLEMENTASI DAN EVALUASI KINERJA DISASTER RECOVERY DALAM LINGKUNGAN DATA CENTER BERBASIS CLOUD

Nama mahasiswa : Yorisan Permana Baginda
NRP : 07111350030014
Pembimbing : 1. Dr.Istas Pratomo, S.T., M.T.
2. Dr. Ir. Achmad Affandi, DEA

ABSTRAK

Meningkatnya pertumbuhan penggunaan internet diimbangi dengan tingginya permintaan pengguna atau penikmat layanan aplikasi agar aplikasi dapat diakses setiap waktu. Di zaman milenial ini, ketersedian aplikasi menjadi raja dari segala sesuatu, jika aplikasi tidak dapat diakses bahkan dalam satu menit akan menjadi masalah besar bagi pemilik bisnis dan mengancam reputasi perusahaan terutama yang membutuhkan ketersediaan aplikasi selama 24 jam contohnya aplikasi perbankan, aplikasi e-commerse, dsb. Untuk menjamin ketersediaan aplikasi dan sistem mayoritas perusahaan yang memiliki infrastruktur tradisional atau private server memiliki disaster planning mulai dari cara paling sederhana yaitu melakukan backup secara berkala maupun membangun sistem Disaster Recovery secara realtime.

Saat ini, ada banyak penyedia cloud yang menawarkan layanan yang disebut DRaaS (Disaster Recovery as a Service) untuk memfasilitasi kebutuhan perusahaan yang ingin menjaga kelangsungan bisnis mereka. Banyaknya penyedia layanan cloud serta miripnya layanan yang ditawarkan membuat calon pengguna bingung penyedia cloud mana yang sesuai dengan sistem bisnis mereka. Penelitian ini memperkenalkan metode desain implementasi DRaaS dari dua penyedia layanan cloud berbeda untuk mendapatkan parameter RPO dan RTO yang nantinya dijadikan pertimbangan dalam memilih layanan DRaaS. Hasil dari penelitian ini adalah sebuah sistem disaster recovery yang menggunakan Google Cloud dan Amazon Web Service sebagai secondary server yang menghasilkan nilai RPO 3 menit. Sistem DRaaS ini telah diuji untuk melakukan failover dengan nilai RTO 9,6 detik untuk Google Cloud dan 15,4 detik untuk Amazon Web Service.

Kata kunci: DRaaS, Amazon Web Service, Google Cloud

Halaman ini sengaja dikosongkan

IMPLEMENTATION AND EVALUATION OF DISASTER RECOVERY PERFORMANCE IN CLOUD-BASED DATA CENTER ENVIRONMENT

By : Yorisan Permana Baginda
Student Identity Number : 07111350030014
Supervisor(s) : 1. Dr.Istas Pratomo, S.T., M.T.
2. Dr. Ir. Achmad Affandi, DEA

ABSTRACT

The increasing growth of Internet use is offset by the high demand of users or connoisseurs of application services for applications to be accessed at any time. In this millennial era, the availability of applications to be the king of everything, if the application can not be accessed even within a minute will be a big problem for business owners and threaten the reputation of the company especially those requiring 24 hours application availability for example banking applications, e-commerce applications, etc. To ensure the availability of applications and systems the majority of companies that have a traditional infrastructure or private server has a disaster planning starting from the simplest way of doing regular backups and build system Disaster Recovery in realtime.

Currently, there are many cloud providers offering services called DRaaS (Disaster Recovery as a Service) to facilitate the needs of companies that want to keep their business going. The number of cloud service providers and similar services offered to make prospective users confused which cloud provider that suits their business systems. This research introduces the design method of DRaaS implementation from two different cloud service providers to get RPO and RTO parameters which will be taken into consideration in choosing DRaaS service. The results of this study is a disaster recovery system that uses Google Cloud and Amazon Web Service as a secondary server that produces a 3 minute RPO value. This DRaaS system has been tested for failover with RTO value 9.6 seconds for Google Cloud and 15.4 seconds for Amazon Web Service.

Key words: DRaaS, Amazon Web Service, Google Cloud

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puja dan puji syukur kepada Allah SWT atas segala rahmat dan karunia yang telah dilimpahkan kepada penulis sehingga penulisan tesis dengan judul :

"IMPLEMENTASI DAN EVALUASI KEINERJA DISASTER RECOVERY DALAM LINGKUNGAN DATA CENTER BERBASIS CLOUD"

Dapat diselesaikan dengan baik. Buku tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar magister pada program studi Teknik Elektro dengan bidang keahlian Telekomukasi Multimedia, Institut Teknologi Sepuluh Nopember.

Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih sedalam-dalamnya kepada pihak-pihak yang membantu terselesainya penelitian ini. Penulis menyadari bahwa dalam penulisan tesis ini masih jauh dari kata sempurna, untuk itu demi perbaikan dan penyempurnaan tesis ini maka saran dan kritik membangun sangat diharapkan. Besar harapan penulis bahwa buku tesis ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa jurusan Teknik Elektro pada khususnya.

Surabaya, 12 Mei 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Munculnya teknologi cloud computing telah mengubah paradigma dunia bisnis dalam menerapkan teknologi dalam proses bisnisnya secara masif. Layanan berbasis cloud sebelumnya, IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*), dan SaaS (*Software as a Service*) telah membuat banyak bisnis yang sebelumnya menggunakan sistem infrastruktur tradisional pindah ke infrastruktur berbasis cloud [1]untuk meningkatkan keandalan dari bisnis mereka. Ketersediaan aplikasi merupakan hal terpenting dalam infrastruktur sistem, jika aplikasi tidak dapat diakses bahkan dalam satu menit akan menjadi masalah besar bagi bisnis dan mengancam reputasi perusahaan terutama yang membutuhkan ketersediaan aplikasi selama 24 jam dalam sehari. *Disaster* atau bisa disebut bencana merupakan salah satu dari banyak faktor yang menyebabkan sistem atau aplikasi tidak tersedia untuk memberikan layanan. Kebakaran, angin topan, banjir, pemadaman listrik, putusnya jaringan telekomunikasi, bahkan kegagalan perangkat keras, bug perangkat lunak, serta kesalahan administrator mengakibatkan *corruption data* atau *deleted object cloud* dapat terjadi dianggap sebagai *Disaster* yang dapat terjadi setiap hari [2]. Jika sistem perusahaan termasuk semua aplikasi dan basis data terpengaruh dan terjadi *downtime*, ini mungkin menjadi masalah substansial bahkan keruntuhan untuk bisnis tersebut.

Perencanaan *recovery disaster* muncul sebagai salah satu dari banyak upaya yang dapat menyelamatkan sistem kita dari bencana yang tidak diinginkan. Amazon Web Service (AWS) [3] mendefinisikan *recovery disaster* adalah segala sesuatu yang berhubungan dengan persiapan dan pemulihan dari *disaster*. Segala sesuatu yang memiliki dampak buruk pada keberlanjutan bisnis perusahaan dapat disebut sebagai *disaster*. Disaster atau bencana termasuk diantaranya kerusakan perangkat keras atau perangkat lunak, pemadaman jaringan, pemadaman listrik, kerusakan fisik pada bangunan seperti kebakaran atau banjir, kesalahan manusia, atau kejadian penting lainnya [4].Setiap perusahaan harus memiliki infrastruktur sendiri untuk mendukung proses bisnis terutama yang membutuhkan permintaan yang besar dari

ketersediaan aplikasi. Oleh karena itu, rencana *recovery disaster* bukan merupakan pilihan tetapi merupakan keharusan.

Penelitian ini akan menggunakan dua metrik untuk menganalisis pengujian *recovery disaster* di antara penyedia cloud atau cloud provider. RTO (*Recovery Time Objective*) dan RPO (*Recovery Point Objective*) adalah dua metrik yang paling penting dalam perencanaan *disaster recovery*[5]. RTO adalah lamanya layanan aplikasi tidak tersedia hingga layanan telah dipulihkan dan dapat memulai layanan aplikasi kembali. Sedangakan , RPO berarti jumlah maksimum data yang dapat hilang ketika pemulihan berhasil dalam satuan waktu.

Saat ini, ada banyak penyedia DRaaS seperti AWS, Google Cloud, OVH yang menawarkan *disaster recovery* dalam layanan mereka [1]. Setiap provider hampir menawarkan fitur yang sama yang membingungkan pelanggan. Oleh karena itu, kami melakukan beberapa pengujian antara penyedia yang berfokus pada nilai RTO dan RPO untuk membantu pelanggan mencari pilihan terbaik untuk perencanaan *disaster recovery* mereka.

1.2 Rumusan Masalah

Banyaknya cloud provider yang menyediakan fitur Disaster Recover as a Service yang menawarkan fitur hampir serupa membuat banyak pengguna sering merasa kebingungan untuk memilih provider mana yang cocok dengan proses bisnis mereka. Pada penelitian yang terdahulu, pengujian benchmark cloud provider hanya sekedar dari segi beban kerja CPU, memory, dan jaringan. Penelitian ini melakukan implementasi sistem Disaster Recovery dari dua buah cloud provider berbeda dan menguji sistem tersebut untuk mendapatkan parameter RTO dan RPO.

1.3 Tujuan

Tujuan dari penelitian ini yaitu melakukan implementasi sistem Disaster Recovery as a Service antara beberapa cloud provider dan melakukan evaluasi kinerja dari sistem DRaaS yang dibangun tersebut dimana dapat diperoleh hasil

perbandingan beserta analisis sehingga dapat menjadi pertimbangan pembuat sistem dalam menentukan provider cloud yang diinginkan.

1.4 Batasan Masalah

Pada tesis ini penulis memberikan batasan permasalahan sebagai berikut :

1. Model layanan cloud yang digunakan yaitu IaaS (Infrastructure as a Service)
2. Layanan yang akan diuji menggunakan adalah web server e-commerce
3. Parameter yang diuji adalah durasi RTO dan RPO
4. Penelitian yang dilakukan tidak membahas dari sisi keamanan (security) sistem
5. Cloud provider yang akan digunakan yaitu Amazon Web Service, Google Cloud, dan Cloud Kilat
6. Sistem operasi yang akan digunakan yaitu Linux Ubuntu 16.04 LTS

1.5 Kontribusi

Kontribusi dari penelitian ini yaitu mengenalkan evaluasi parameter baru dalam pengujian infrastruktur Disaster Recovery as a Service sehingga dapat menjadi tambahan pertimbangan dalam pemilihan cloud provider yang menyediakan layanan DRaaS.

1.6 Metodologi Penelitian

Metodologi penelitian yang dilakukan dalam penelitian ini dapat dijabarkan secara detail sebagai berikut :

1. Studi Literatur

   Studi literatur dilakukan untuk mengumpulkan informasi dari literatur – literatur seperti paper, jurnal, buku dan literatur lainnya. Informasi yang diperoleh digunakan untuk memecahkan masalah yang telah dirumuskan dalam tesis ini. Adapun literatur yang dipelajari adalah yang berkaitan dengan *disaster*, *disaster recovery*, *disaster recovery plan, Virtual machine* (VM) dan *Cloud Computing.*

2. Desain System

Setelah melakukan studi literatur yang dibutuhan, langkah selanjutnya yaitu mulai melakukan desain infrastruktur sistem yang akan diimplementasikan. Dalam tahap ini perencanaan infrastruktur mulai dibangun dari mulai pemilihan cloud provider, layanan dan tools yang akan digunakan seperti tools untuk replikasi dan failover dsb

3. Pengujian Sistem

   Pada tahap pengujian, infrastruktur DRaaS yang telah dibangun akan dilakukan pengujian berdasarkan skenario yang dibuat. Setelah itu akan dilakukan analisis berdasarkan metrik yang telah didapatkan dari hasil pengujian sistem

4. Penarikan Kesimpulan

   Tahap paling akhir dari penelitian ini adalah membuat kesimpulan dari hasil analisis data yang telah didapatkan dari pengujian untuk memberikan jawaban terhadap permasalahan penelitian serta memberikan saran untuk perbaikan penelitian berikutnya.

5. Penyusunan Laporan Tesis

   Di tahap akhir ini akan dilakukan penyusunan laporan dari hasil penelitian yang diperoleh dengan mengikuti format penulisan tesis yang telah diberikan.

BAB 2
KAJIAN PUSTAKA

2.1 Disaster Recovery

Disaster Recovery merupakan seluruh usaha yang dilakukan untuk mempersiapkan dan upaya pemulihan jika terjadi bencana[6]. Disaster atau bencana merupakan segala macam kejadian yang mempunyai efek buruk bagi keberlangsungan bisnis perusahaan diantaranya termasuk kegagalan hardware atau software, pemutusan jaringan maupun daya listrik, kesalahan manusia, maupun bencana fisik sebenarnya seperti kebakaran, banjir, dsb.

Gregory[7] mendefinisikan beberapa jenis bencana yang dapat terjadi pada infrastruktur sistem suatu perusahaan yaitu :

1. Kebakaran
2. Banjir
3. Tornado
4. Angin topan
5. Badai angin dan es
6. Badai parah
7. Kebakaran hutan
8. Tanah longsor
9. Longsor
10. Tsunami
11. Gempa bumi
12. Letusan gunung berapi
13. Masalah keamanan
14. Kegagalan perangkat
15. Kegagalan daya listrik
16. Kegagalan utilitas
17. Arson
18. Pandemik

19. Sabotase
20. Pemogokan dan pemberhentian kerja
21. Pemadaman
22. Gangguan warga sipil
23. Terorisme
24. Perang

Manfaat dari Disaster Recovery Planning[1] yaitu :

1. Peningkatan proses bisnis
2. Peningkatan teknologi yang membuat sistem IT menjadi lebih konsisten dan lebih terkontrol dan terawasi
3. Lebih sedikit gangguan karena sistem lebih stabil. Gangguan-gangguan yang sebelumnya sering terjadi menjadi berkurang dan cepat teratasi
4. Kualitas layanan yang lebih tinggi karena meningkatnya perbaikan proses dan teknologi.
5. Semakin kompetitif dibandingkan dengan kompetitor karena tingginya ketersediaan layanan dan kehandalan yang meningkat dibanding kompetitor.

Terdapat beberapa pendekatan pembangunan disaster recovery[7] yaitu :

1. Hot Site

Pendekatan Hot Site merupakan salah satu pendekatan pembangunan Disaster Recovery dimana organsasi atau perusaan melakukan duplikasi seluruh infrastruktur sistem termasuk server, penyimpanan data, dan jaringan dimana lokasi dari duplikat infrastruktur berbeda dengan infrastruktur aslinya. Pendekatan jenis ini membutuhkan biaya yang sangat banyak karena proses duplikasi diimplementasikan mulai dari hardware sampai ke software.

Terdapat 2 mode dalam pendekatan Hot Site yaitu active-active dan active-passive. Pada Active-Active mode, aliran proses update data menjadi 2 arah, dari infrastruktur asli ke infrastruktur backup dan sebaliknya saat terjadi kegagalan

sistem. Sedangkan pada mode active-passive arah update data hanya berlangsung dalam 1 arah. Yaitu dari infrastruktur asli ke infrastruktur backup.

2. Cold Site

Pendekatan Cold Site menggunakan teknik backup secara berkala yang dilakukan pada infratruktur utama. Saat proses backup data dikirim ke lokasi lain yang berbeda dengan lokasi infrastruktur utama. Pada

3. Warm Site

Pendekatan Warm Site merupakan jalan tengah dari 2 pendekatan sebelumnya dengan menggabungkan teknik Hot dan Cold. Perusahaan dapat mengimplementasikan teknik Warm Site jika ingin mengadaptasi teknik Hot site tetapi tidak mempunyai biaya yang besar dengan konsekuensi proses penanganan jika terjadi bencana tidak secepat Hot Site.

2.2 Disaster Recovery as a Service (DRaaS)

Disaster Recovery as a Service (DRaaS) merupakan layanan cloud yang muncul setelah 3 jenis layanan cloud sebelumnya yaitu Infrastructur as a Service (IaaS), Platform as a Service (PaaS) serta Sofware as a Service (SaaS) yang telah sukses menarik minat banyak pengguna untuk memindahkan infrastruktur lama nya ke infrastruktur berbasis cloud[8]. DRaaS merupakan layanan berbasis cloud yang memudahkan organisasi atau perusahaan mengatur dan mengelola sistem disaster recovery yang dimiliki secara cepat dengan tingkat fleksibilitas yang tinggi

Terdapat 2 jenis DRaaS[8] yaitu :

1. Public Cloud DRaaS
2. Private Cloud DRaaS

2.3 Data Center

Data center adalah suatu fasilitas berupa ruangan fisik yang digunakan untuk menempatkan beberapa server dan komponen-komponen terkaitnya, seperti piranti jaringan komputer dan penyimpanan data dalam sebuah lokasi yang sama[3].

Fasilitas ini biasanya mencakup juga catu daya redundan atau cadangan, koneksi komunikasi data redundan, pengontrol lingkungan (mis. AC, Ventilasi), pencegah bahaya kebakaran, serta piranti keamanan fisik.

2.4 Cloud Computing

Cloud computing merupakan tren baru dalam bidang teknologi informasi yang memindahkan komputasi dan penyimpanan data dari perangkat komputer fisik ke *data center* yang besar . Barry Sosinky dalam bukunya [9] mendefinisikan *cloud computing* sebagai kumpulan aplikasi dan layanan yang berjalan dan beroperasi dalam jaringan terdistribusi atau tersebar dengan memanfaatkan sumber daya *virtual* dan standar *internet*. Sejarah *cloud computing* berawal dari layanan baru perusahaan telekomunikasi yang menawarkan produk VPN (Virtual Private Network) yang sebelumnya merupakan sirkuit data yang bersifat point-to-point. Kekurangan layanan point-to-point ini adalah mahal dan boros *bandwidth*, sehingga muncul teknologi VPN sebagai layanan alternatif dimana kualitas yang ditawarkan sama dengan *point-to-point* tetapi jauh lebih murah jika ditinjau dari segi biaya.

Terdapat beberapa manfaat yang dapat diperoleh dalam penggunaan *cloud computing*[10] :

1. Penggunaan *cloud computing* dapat menekan biaya operasional karena sistem beroperasi dengan efisiensi dan pemanfaatan sumber daya yang lebih besar.
2. Tidak memerlukan lisensi *software* dan *hardware* untuk membangun sistem yang diinginkan.
3. Adanya jaminan QOS (Quality of Service) dari penyedia layanan *cloud computing.*
4. Beberapa fitur seperti *load balancing* dan *failover* menjamin kehandalan dalam layanan *cloud computing*
5. Dengan adanya pihak lain yang mengelola infrastruktur sistem dalam *cloud*, perusahaan yang menyewa layanan *cloud computing* dapat fokus ke pengelolaan bisnis yang berjalan.

6. Pemeliharaan sistem dan infrastruktur menjadi lebih sederhana dan mudah karena sistem tersentralisasi

Selain berbagai macam keuntungan diatas, penggunaan layanan *cloud computing* juga memiliki beberapa kekurangan [9] yaitu :

1. Meskipun layanan *cloud computing* mendukung elastisitas sesuai proses bisnis dan keinginan pengguna, tetapi tidak semua keinginan pengguna akan dilayani dan diimplementasikan.
2. Seluruh aplikasi *cloud computing* memiliki masalah yang sama dalam hal koneksi jaringan internet khususnya latensi. Model *cloud computing* tidak cocok ketika aplikasi yang digunakan oleh pengguna membutuhkan transfer data dalam jumlah besar.

2.4.1 Amazon Web Service DRaaS

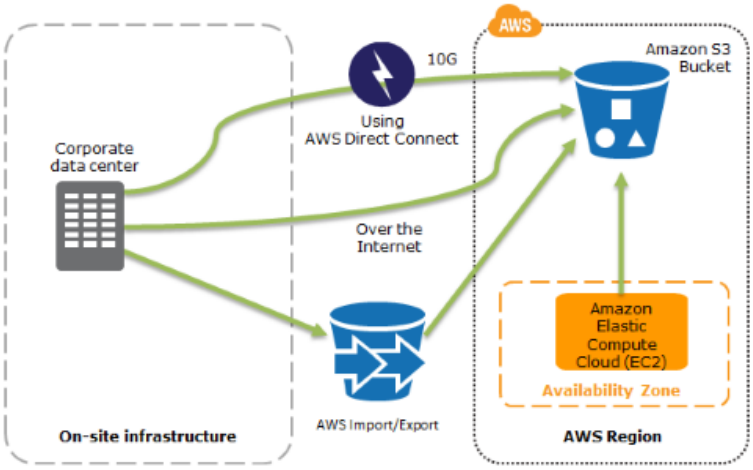
Amazon Web Service (AWS) [11] merupakan salah satu penyedia layanan infrastruktur *cloud computing* komersial terintegrasi. AWS medukung skema DRaaS dengan menyediakan beberapa fasilitas pembangunan DRaaS yaitu :

1. Regions
2. Strorage
3. Gateway
4. Compute Engine
5. Networking
6. Database
7. Security and Compliance

AWS menyediakan beberapa opsi infrastruktur menyesuaikan dengan kebutuhan sistem dari client. Gambar 2.1 mendeskripsikan spektrum pembangunan DRaaS dalam AWS. Terdapat 4 opsi pilihan yaitu :

1. Backup and Restore

Merupakan opsi paling sederhana dari seluruh skema DRaaS yang ditawarkan oleh AWS. Dalam skema Backup and Restore, data dari infrastruktur client dapat dibackup ke AWS melalui berbagai cara seperti transfer image sistem yang berisi sistem operasi ke Amazon S3, layanan penyimpanan terintegrasi yang dimiliki oleh AWS.

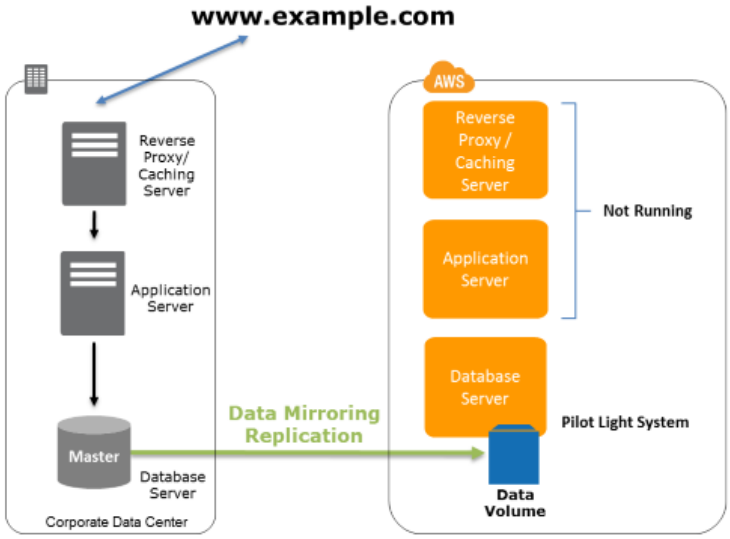


Gambar 2.1 Opsi Backup & Restore AWS

2. Pilot Light

Pilot light merupakan skenario DRaaS yang mirip dengan versi Backup and Restore tetapi dengan menggunakan versi minimal infrastruktur yang berjalan di cloud AWS. Pilot light memungkinkan client memiliki duplikat infrastruktur di sistem AWS yang akan mengambil alih layanan apabila terjadi disaster.

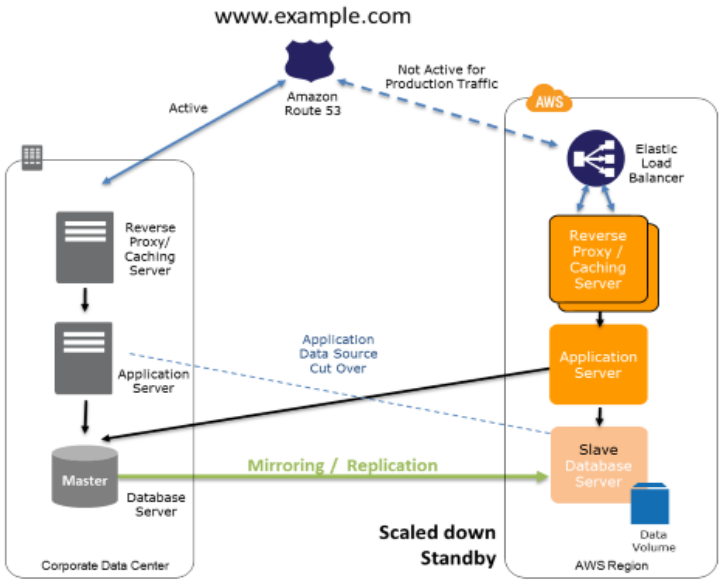
Ketika kondisi normal layanan akan diarahkan ke infrastrktur client, dan infrastruktur di AWS dalam posisi offline. Konsistensi data sistem antara 2 infrastruktur akan dihandle oleh data mirroring replication. Saat disaster terjadi dan infrastruktur utama client dalam posisi offline, maka Elastic Load Balancer yang akan bertugas mengalihkan layanan ke Infrastruktur AWS sehingga layanan masih dapat berjalan.



Gambar 2.2 Opsi Pilot Light AWS

3. Warm Standby

Dalam opsi Warm Standby, infrastruktur client juga memiliki duplikat infrastruktur di cloud AWS dan juga akan menangani peralihan layanan dari infrastruktur client jika terjadi failover. Perbedaan dari Pilot Light adalah jika Pilot Light merupakan versi minimal version dari infrastruktur AWS, maka Warm Standby merupakan sistem versi lengkap dari infrastruktur AWS.

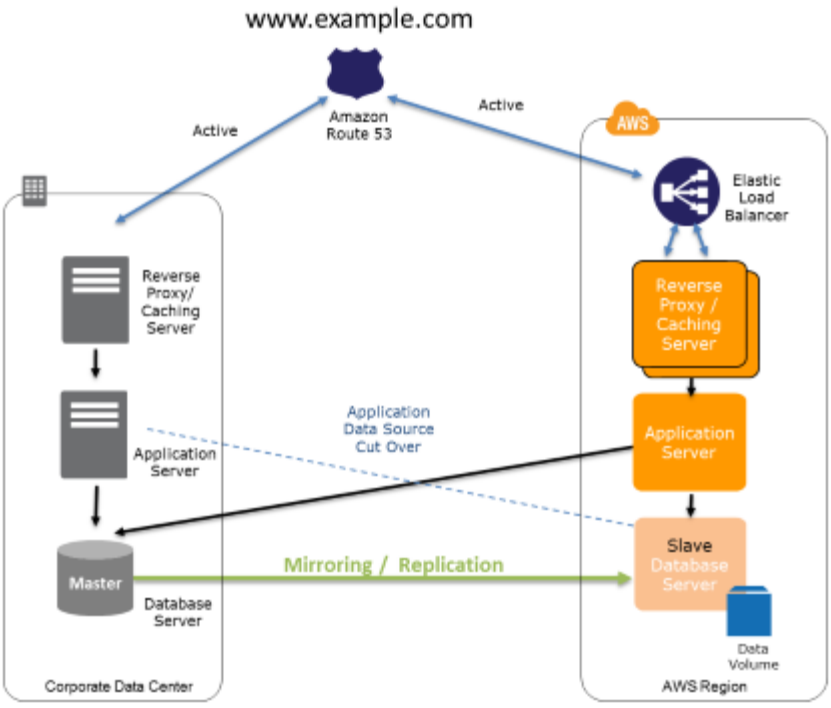


Gambar 2.3 Opsi Warm Standby AWS

Selain itu jika pada Pilot Light, saat kondisi normal status sistem dari infrastruktur duplikat yang ada di AWS berstatus offline, maka pada Warm Standby sistem di AWS dalam kondisi Standby dimana nantinya pada saat failover, skenario Warm Stndby kan lebih cepat dalam proses recovery layanan.

4. Multi-Site Solution Deployed on AWS and On-Site

Multi-Site merupakan skenario paling ideal dari seluruh opsi yang ditawarkan AWS dan cocok diimpementasikan untuk sistem real-time dengan tingkat kebutuhan recovery yang cepat.



Gambar 2.4 Opsi Multi-Site AWS

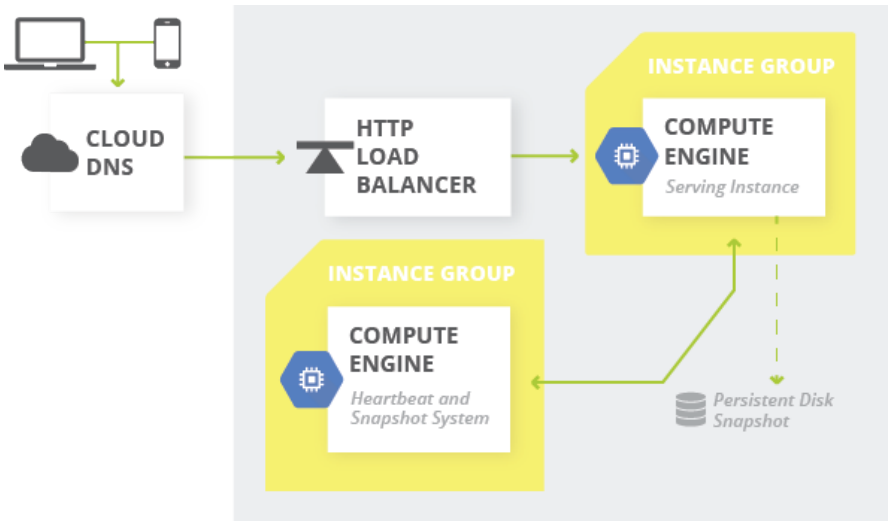
Dibandingkan dengan Warm Standby, proses Multi-Site akan menghasilkan proses recovery yang lebih cepat karena saat kondisi normal, duplikat infrastruktur di sistem AWS dalam kondisi Active dan siap menerima peralihan layanan setiap waktu.

2.4.2 Google Cloud Platform

Google Cloud Engine merupakan salah satu jenis layanan Google Cloud dimana menyediakan private atau public server untuk berbagai kebutuhan. Google

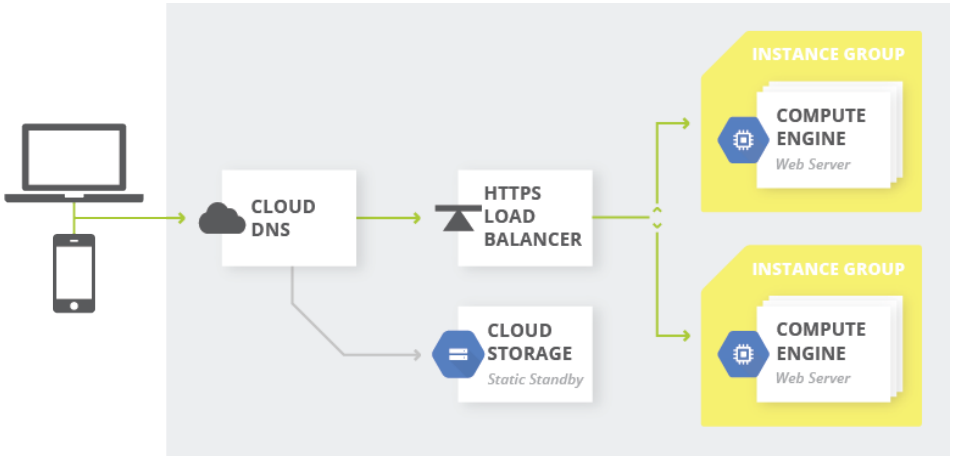
Cloud merupakan salah satu cloud provider komersial yang menyediakan layanan cloud terlengkap mulai dai IaaS (Infrastructure as a Service), PaaS (Platform as a Service), sampai SaaS (Software as a Service)[12].

Google Cloud menyediakan opsi layanan infrastruktur untuk client apabila ingin mengimplementasikan DRaaS menggunakan layanan Google Cloud yaitu Cold Standby, Warm Standby, dan Remote Recovery[3]. Berbagai opsi tersebut didukung oleh layanan fitur-fitur dari Google Cloud mulai dari GCE (Google Compute Engine) untuk pembuatan Virtual Private Server, Cloud Storage, CloudSQL, BigQuery untuk data backup dan recovery, HTTP load balancing dan CloudDNS untuk penangan failover dan failback, serta pemantauan dan monitoring sistem melalui Google Cloud Deployment Manager.



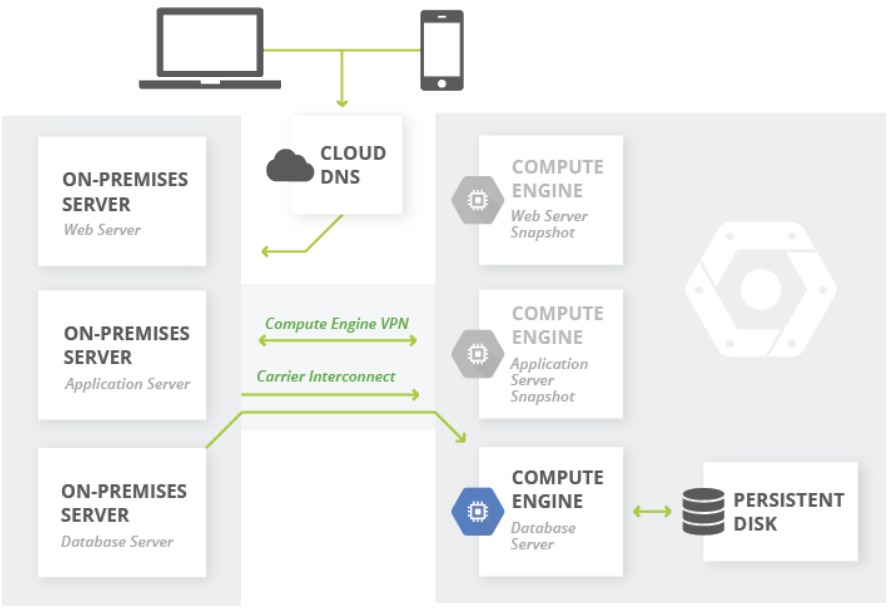
Gambar 2.5 Opsi Cold Standby Google Cloud[3]

Gambar 2.2 menunjukkan diagram skema dari opsi Cold Standby Google Cloud. Dalam skema Cold Standby terdapat 2 server identik masing-masing bertugas sebagai main server dan standby server. Ketika main server tidak bisa diakses, standby server yang bertugas menggantikan tugas pelayanan aplikasi. Sedangkan untuk sistem backup data pada skema ini menggunakan snapshot persistent disk secara periodik dari main server ke standby server. Persistent disk merupakan penyimpanan khusus/storage disk terintegrasi yang berisi sistem operasi yang digunakan Google Cloud Engine. Proses pengalihan layanan ketika terjadi diaster menggunakan HTTP Load Balancer dan Heartbeat.



Gambar 2.6 Opsi Warm Stanby Google Cloud

Selain Cold Standby, Google Cloud juga menyediakan opsi Warm Standby yang diilustrasikan oleh Gambar 2.3 dimana terdapat 2 backend server yang melayani permintaan web server oleh client. Pada kondisi normal Cloud DNS akan mengarahkan layanan aplikasi ke primary aplication dalam Compute Engine Group, sedangkan saat terjadi offline maka CloudDNS akan mengarahkan ke Static Standby Cloud Storage.



Gambar 2.7 Opsi Remote Recovery Google Cloud

Google Cloud menyediakan opsi pilihan lain kepada pelanggan yang ingin menggunakan layanan DRaaS Google dengan menggabungkan infrastruktur sistem

lama/on-premises dengan sistem Google Cloud seperti terlihat pada Gambar 2.4. Pelanggan dapat menghubungkan jaringan dari infrastruktur lama ke infrastruktur Google Cloud dengan VPN atau Carier Interconnect untuk keperluan backup data.

2.4.3 CloudKilat

Cloudkilat merupakan salah cloud provider yang terletak di Indonesia dengan harga yang cukup terjangkau dibandingkan dengan cloud provider Indonesia lain. Diluncurkan pertama kali pada bulan September 2013, CloudKilat berfokus menyediakan layanan komputasi terdistribusi bagi kalangan usaha kecil menengah atau startup di Indonesia[13]. Pusat data Cloudkilat terletak di IDC Duren 3, Jakarta Selatan, yang merupakan jantung dari peering Internet Indonesia yang terbesar yaitu OpenIXP .

2.5 Replikasi MySQL

Replikasi MySQL adalah fitur dari MySQL Server yang memungkinkan untuk menduplikasi/mereplikasi data dari satu server database MySQL(Master) ke satu atau lebih database MySQL lain(Slaves). MySQL Replication merupakan teknologi yang fleksibel dan kuat juga telah didukung di MySQL untuk waktu yang sangat lama. MySQL Replication dapat digunakan untuk mereplikasi semua database, beberapa database, bahkan hanya beberapa tabel yang terdapat dalam satu database[14].

Server yang terlibat dalam sistem replikasi memiliki salah satu dari dua peran yaitu :

1. Master : Server master menulis semua transaksi yang mengubah data ke log biner

2. Slave : Server slave terhubung dengan server master dan mengambil transaksi dari log binary server master lalu menerapkannya perubahannya ke server lokal. Slave dapat bertindak juga sebagai master

Replikasi bekerja kerena event yang ditulis ke dalam binary log pada master , dibaca dan diproses oleh slave. Event-event tersebut dicatat dalam binary log dalam format yang berbeda-beda sesuai tipenya [14].

MySQL mendukung dua jenis replikasi yaitu statement-based replication dan rowbased replication :

1. Statement Based Replication Tipe replikasi default dari MySQL yang berbasis propagasi statement SQL dari master ke slave. Tipe replikasi jenis ini tersedia sejak versi MySQL 3.23 dan memiliki banyak kelebihan, diantaranya space yang dibutuhkan untuk menyimpan file log jauh lebih kecil karena data yang ditulis ke dalam log lebih sedikit meskipun statement update atau delete mempengaruhi banyak row. Kekurangan dari Statement Based Replication adalah tidak semua statement query yang merubah data dapat direplikasi seperti query DELETE atau REPLACE. Statement query yang menggunakan fungsi-fungsi di bawah ini tidak dapat direplikasi oleh Statement Based Replication :

a. LOAD_FILE()

b. UUID(), UUID_SHORT()

c. USER()

d. FOUND_ROWS()

e. SYSDATE()

f. GET_LOCK()

g. IS_FREE_LOCK()

h. IS_USED_LOCK()

i. MASTER_POS_WAIT()

j. RAND() k. RELEASE_LOCK() l. SLEEP() m. VERSION()

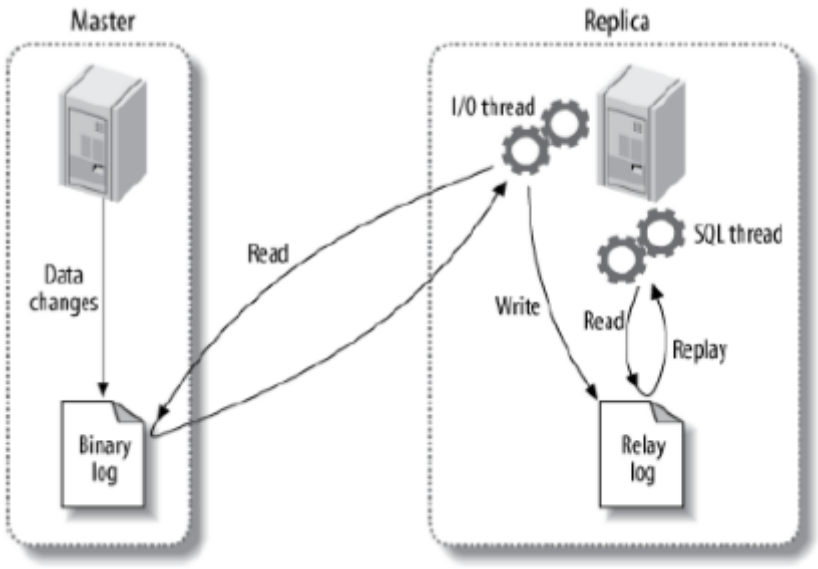
2. Row Based Replication Tipe replikasi dimana master menulis event ke dalam binary log yang menunjukkan individual table rows yang berubah. Kelebihan dari tipe replikasi Row Based Replication yaitu semua perubahan dapat direplikasi, meskipun demikian RBR cenderung menghasilkan lebih banyak data untuk ditulis ke dalam binary log. Jika Statement Based Replication hanya menulis statement perubahan saja ke dalam binary log ketika proses replikasi, Row Based Replication menulis setiap perubahan row ke dalam log.

Cara kerja sederhana dari replikasi MySQL yaitu :

1. Server master mencatat seluruh perubahan ke dalam binary logs

2. Server replika menyalin isi binary log server master ke dalam relay log miliknya

3. Server replika melakukan perubahan dalam datanya mengikuti relay log



Gambar 2.8 Cara Kerja Replikasi

Pada Gambar 2.8 menjelaskan cara kerja replikasi secara lengkap pada masing-masing server yaitu :

1. Server Master Server master melakukan penulisan event/perubahan di log tertentu yaitu binary log. Di dalam binary log terdapat data-data yang nantinya akan dibaca oleh server slave.

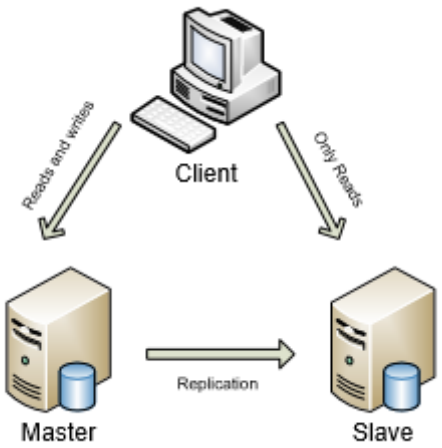
2. Server Slave Pada slave terdapat 2 thread yang berjalan yaitu :

a. I/O Thread : Bertugas untuk terhubung dengan server master, bertanya apakah ada transaksi baru, dan menyalin data dari binary log ke dalam relay log.

b. SQL Thread : Bertugas membaca seluruh transaksi yang terdapat pada relay log dan mengeksekusi nya ke database. SQL Thread menangani bagian terakhir dari proses replikasi yaitu membaca dan menjawab event di relay log.

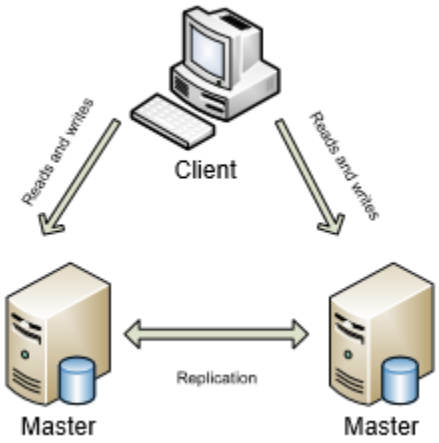
Dalam MySQL terdapat beberapa jenis replikasi, yaitu:

1. Master to Slave Replikasi Master to Slave adalah teknik replikasi yang bersifat satu arah yang terdiri dari minimal 2 buah komputer, yaitu satu buah node master dan satu buah node slave. Pada Gambar 2.9 terlihat dalam teknik replikasi Master to Slave, yang bisa melakukan penulisan dan pembacaan database hanya node master saja. Node slave hanya bisa melakukan pembacaan data.



Gambar 2.9 Replikasi Master to Slave

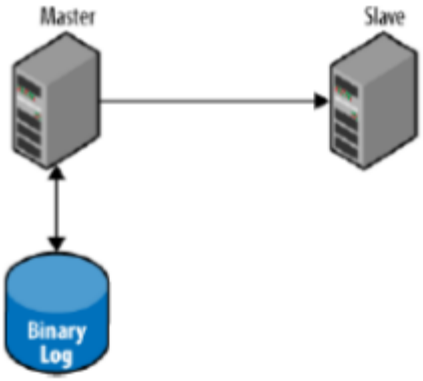
2. Master to Master Replikasi Master to Master adalah teknik replikasi yang bersifat dua arah, berbeda dengan Master to Slave yang hanya satu arah. Jadi kedua server dapat melakukan pembacaan dan penulisan data dalam sistem seperti terlihat pada Gambar 2.10. Terdiri dari minimal dua komputer yaitu satu node master utama dan satu node master backup.



Gambar 2.10 Replikasi Master to Master

Terdapat dua jenis tipe replikasi Master to Master/Dual Master yaitu Active-Active dan Active-Passive. Dalam replikasi Active-Active, semua server master dapat saling mengupdate database satu sama lain dalam waktu yang bersamaan. Berbeda dengan tipe active-passive, terdapat server yang bertindak sebagai active master dimana menangani update perubahan dari client sementara passive master hanya mengikuti perubahan yang terjadi di active master. Ketika active master down/fails peran dari kedua server akan berubah, layanan akan diarahkan ke passive server yang akan berubah menjadi active master yang berfungsi menangani update dari client sedangkan active master yang pertama berubah menjadi passive server.

Binary logs merupakan file biner yang berisi rincian dari setiap transaksi eksekusi MySQL server. MySQL Server menciptakan log biner dalam bentuk name.000001, name.000002, dan seterusnya. Setelah log biner mencapai ukuran/size yang ditentukan, log baru akan tercipta dan setelah jangka waktu tertentu MySQL akan menghapus log-log lama tersebut[14].



Gambar 2.11 Cara Kerja Binary Log

Pada Gambar 2.11 binary log membuat replikasi database dapat berjalan, karena binary log mencatat seluruh perubahan yang terjadi pada database server dan kemudian binary log tersebut akan dibaca oleh database server lainnya untuk memulai proses replikasi. Tujuan dari binary log adalah mencatat perubahan yang terjadi pada tabel di dalam database. Binary log hanya berisi perubahan yang terjadi pada database, bukan merubah data pada database

2.6 Load Balancer

Load Balancer merupakan salah satu tools yang digunakan dalam infrastruktur Disaster Recovery dimana bertugas sebagai pemantau aktifitas dari dua server[2]. Penelitian ini menggunakan tools HA-Proxy yang bersifat open source yang bertugas memantau aktifitas primary server dan secondary server agar dapat selalu real-time dalam menangani kondisi disaster.

2.7 Penelitian Terdahulu

Pada penelitian yang dilakukan oleh Hossam[15] merumuskan metode pengembangan IT Disaster Recovery Plan yang membagi proses pembangunan DRaaS/Recovery Strategy menjadi 3 fase besar yaitu :

1. Functional Team and Responsibilities
2. Disaster Recovery Action Plan
3. Evaluating and Testing the Disaster Recovery Plan

Tabel 2.1 menjelaskan subfase dari masing-masing fase pada disaster recovery strategi yang disulkan oleh Hossam.

Sindhura[5] mendefinisikan beberapa parameter performansi dalam pembangunan implementasi Disaster Recovery as a Service, diantaranya yaitu :

Tabel 2.1 Disaster Recovery Phases[15]

Phase I Functional Teams and Responsibilities	Phase II Disaster Recovery Action Plan	Phase III Evaluating and Testing the Disaster Recovery plan
• Damage Assessment Team • Disaster Recovery Team • Restoration Team • Operations Team • Customer Support Team • Major Plan Components - format and structure	• Backup and off-site storage procedures • Backup Facility • Disaster Preparation • Emergency Response • Recovery Procedures • Recovery Time Table	• Testing the Disaster Recovery Plan • Hot Site (DR Site) Test Procedures • Hot Site (DR Site) Test Planning • Application Testing Support • Post-Test Wrap-Up • Hot Site (DR Site) Test Schedule • Maintaining the Plan

1. Recovery Point Objective(RPO) dan Recovery Time Objective(RTO)

RTO (Recovery Time Objective) merupakan waktu yang dibutuhkan sistem untuk kembali mengaktifkan layanan yang mati setelah bencana terjadi. Nilai RTO bergantung pada sejumlah perintah yang dibutuhkan untuk mengembalikan pengaturan transaksi yang terjadi pada backup atau secondary server. Contoh penerapan dari RTO yaitu jika bencana terjadi pukul 12.00 siang dan nilai RTO yaitu 8 jam, maka sistem harus dapat mengaktifkan layanan yang mati pada pukul 20.00 malam. Dalam sistem yang menggunakan teknologi tape-based dimana masih menggunakan cara tradisional dalam proses backup data waktu yang dibutuhkan dapat berhari-hari, berbeda dengan sistem replikasi secara real-time yang membutuhkan waktu jauh lebih singkat bahkan dalam satuan menit.

Sedangkan RPO (Recovery Point Objective) merupakan jumlah besar data hilang yang diizinkan setelah terjadi bencana dalam satuan waktu. Contoh penerapannya yaitu jika bencana terjadi pukul 12.00 siang dan nilai RPO yaitu 1 jam, maka sistem harus bisa melakukan recovery atau perbaikan data yang ada pada sistem sebelumnya yaitu sebelum pukul 11.00 siang

Suguna et al [16] memetakan rentang nilai RTO dan RPO berdasarkan tier/lapisan jenis backup.

Tabel 2.2 Pemetaan Nilai RTO dan RPO

Tier	Description	RTO	RPO
1	Point in time tape backup	2-7 days	2-24 hours
2	Tape backup to remote site	1-3 days	2-24 hours
3	Disk point in time copy	2-24 hours	2-24 hours
4	Remote logging	12-24 hours	5-30 min
5	Concurrent ReEx	1-12 hours	2-10 min
6	Mirrored data	1-4 hours	0-5 min
7	Mirrored data with failover	0-60 min	0-5 min

Pada Tabel 2.2 memetakan nilai RTO dan RPO dengan jenis tier sistem Disaster Recovery yang dibangun. Rentang nilai yang didefinisikan menjelaskan batas minimum dan maksimum nilai yang dapat digunakan acuan ketika akan mengevaluasi sistem Disaster Recovery yang telah dibangun. Contoh untuk tier 7 yaitu sistem Disaster Recovery yang mengimplementasikan sistem mirror replikasi data dengan teknik failover membutuhkan waktu dengan rentang 0-60 menit untuk nilai RTO dan 0-5 menit untuk nilai RPO.

2. Age of Backup atau Lama waktu backup mendefinisikan periode waktu dalam jam atau hari yang digunakan untuk melakukan backup atau duplikasi data.

3. Time Taken to Backup merupakan waktu yang dibutuhkan oleh proses backup untuk menyelesaikan operasinya sampai selesai dan data telah sukses sampai pada tujuan.
4. Time Taken to Recover merupakan waktu yang digunakan server untuk mengembalikan data yang hilang dimana bergantung pada kecepatan jaringan, besarnya data, jenis jaringan, kecepatan jalur dari media transmisi dsb.
5. Total Cost of Ownership yaitu perkiraan jumlah biaya yang dibutuhkan untuk pembangunan Disaster of Recovery.
6. CPU Utilizations merupakan parameter uji untuk memantau behaviour dari load CPU dari server backup dalam sistem disaster recovery.
7. Memory Utilizations meruapakan parameter uji untuk memantau behaviour dari load memory dari server backup dalam sistem disaster recovery.

Seluruh metrik parameter diatas telah digunakan oleh Sindhura[5] dalam penelitiannya dimana membandingkan efektifitas backup antara backup tradisional (tape-based) dengan backup menggunakan cloud (cloud-based).

Peter [7] mendefinisikan beberapa hal penting yang harus diperhatikan sebelum proses pengujian pada sistem Disaster Recovery, yaitu :

1. Practice First, dimana melakukan pengujian virtual dengan prosedur yang akan dipakai saat melakukan pengujian.
2. Workload, merupakan salah satu parameter penting dalam pengujian sistem, karena sistem disaster recovery yang dibangun harus dapat menangani keseluruhan workload atau beban. Dalam parameter workload, stress testing atau volume testing dapat diimplementasikan untuk menguji sistem recovery agar sesuai harapan yang diinginkan. Salah satu contohnya yaitu pengujian akses oleh banyak user pada aplikasi web.
3. Reach-ability, merupakan satu kondisi dimana seluruh user dalam aplikasi termasuk semua akses seperti customer, user, supplier, admin dll harus

dapat mengetahui bagaimana mengakses sistem aplikasi ketika terjadi pengujian disaster.

4. Notification and communication, merupakan salah satu cara untuk mengkomunikasikan kepada seluruh user ketika terjadi pengujian Disaster Recovery

5. Transaction Integrity, merupakan salah satu komponen terpenting dimana integritas data dalam proses transaksi aplikasi merupakan hal wajib yang harus dipertahankan selama terjadinya disaster recovery, baik selama kondisi normal, dan selama disaster atau bencana terjadi.

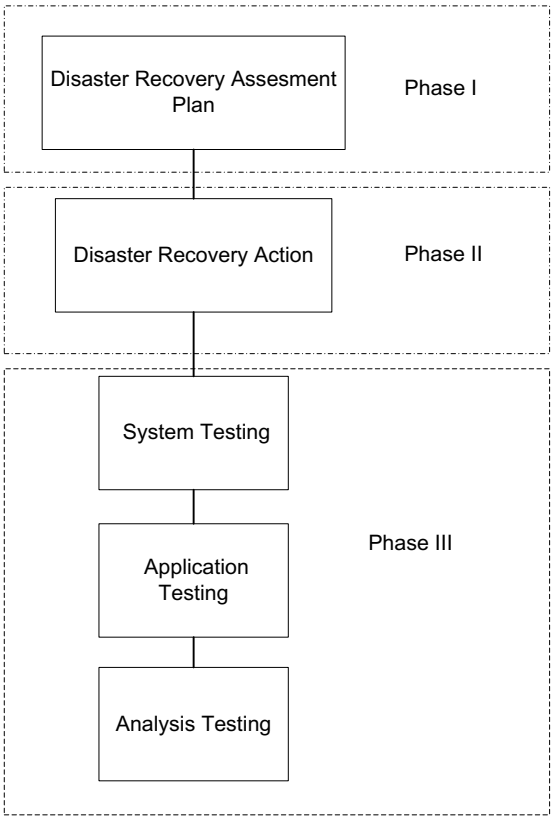
6. Transaction source, dimana aplikasi dapat mendeteksi server bagian mana yang menangani transaksi dan data.

7. Controls, regulation, audits, and security.

BAB 3
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metode yang digunakan dalam penelitian ini mengadaptasi fase *Hossam's data centre recovery* [15], dimana membagi proses pembangunan disaster recovery menjadi 3 fase besar yaitu, Disaster Recovery Assesment Plan, Disaster Recovery Action dan yang terakhir adalah fase 3 dimana terdapat beberapa subfase yaitu System Testing, Application Testing serta Analysis Testing.



Gambar 3.1 Metode Penelitian

Gambar 3.1 menggambarkan metode pengerjaan penelitian ini. Dalam fase 1 terdapat tahap Disaster Recovery Assesment, dimana dalam tahap ini proses penyusunan rencana disaster recovery dari mulai observasi sistem yang telah ada atau existing system salah satunya menganalisa kekurangan sistem disaster

recovery yang lama dimana masih menggunakan metode backup sistem secara periodik, kemudian melakukan perancangan disaster recovery baru yang akan dibangun. Dalam melakukan perancangan sistem yang baru terdapat beberapa poin yang dipertimbangkan dalam tahap ini diantaranya :

1. Jenis aplikasi dan database yang akan dijadikan objek disaster recovery
2. Menentukan mastriks nilai yang akan digunakan dalam mengevaluasi kinerja sistem disaster recovery yang dibangun
3. Menetukan provider cloud yang memenuhi kriteria evaluasi
4. Membuat desain perancangan sistem antara primary cloud dan secondary cloud.
5. Merancang skenario pengujian sistem ketika sistem Disaster Recovery telah terbangun dan berfungsi secondary cloud,

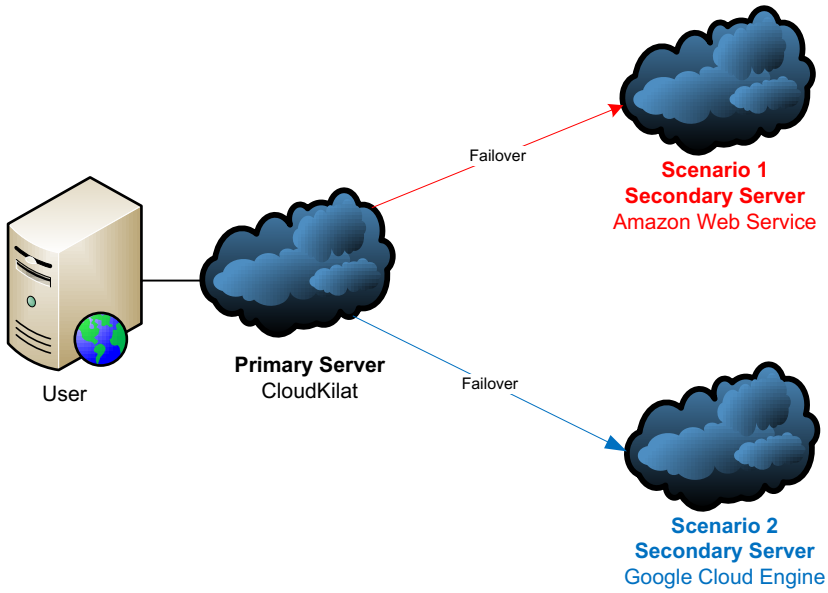
Fase kedua yaitu Disaster Recovery Action dimana semua desain yang telah disusun di tahap sebelumnya akan diimplementasikan seperti membuat duplikat system di secondary cloud, memastikan kerja sistem operasi dan aplikasi berjalan di kedua cloud sesuai fungsinya, mengimplementasikan prosedur replikasi data dan failover antara kedua cloud. Fase ketiga terdiri dari beberapa subfase yaitu System Testing, Application Testing, dan Analysis. Pada System Testing pengujian yang dilakukan yaitu menguji mekanisme failover sistem antara dua cloud provider diantaranya menguji load balancer IP, menguji mekanisme recovery sistem ketika failover sedang berlangsung. Application Testing menguji ketersedian dan kehandalan aplikasi ketika sebelum dan sesudah terjadi failover, dalam subfase ini pengujian konsistensi data juga akan dilakukan. Setelah pengujian sistem yang dibangun selesai dilakukan, tahap selanjutnya yaitu melakukan analisis parameter yang telah ditentukan untuk melakukan evaluasi sistem disaster recovery.

3.2 Rancangan Sistem

Rancangan sistem yang akan dijelaskan dalam bab ini terdiri dari 2 yaitu, rancangan desain infrastruktur sistem dan rancangan aplikasi sistem.

3.1.1 Desain Infrastruktur Sistem

Dalam penelitian ini sistem Disaster Recovery as a Service yang dibangun terbagi menjadi 2 skenario, skenario pertama terdiri dari CloudKilat sebagai primary server dan Amazon Web Service sebagai secondary server, sedangkan untuk skenario kedua ini dalam pengujiannya memakai 3 cloud provider yang berbeda yaitu CloudKilat dimana berlokasi di Indonesia, serta Google Cloud Engine dan Amazon Web Service berlokasi di Amerika Serikat.



Gambar 3.2 Desain Skenario Infrastruktur DRaaS

Pada Gambar menunjukkan desain infrastruktur sistem yang dibangun dalam penelitian ini. Pada skenario pertama user akan mengakses aplikasi website yang terletak di Cloudkilat yang bertindak sebagai primary server, ketika terjadi disaster salah satu contohnya yaitu sistem crash yang mengakibatkan system tidak dapat menjalankan layanan aplikasi pada primary server maka sistem akan secara otomatis mengalihkan layanan aplikasi pada secondary server yaitu server Amazon Web Service. Dalam hal ini user tidak akan pernah tahu bahwa letak server untuk melayani permintaan aplikasi client telah berubah.

Sedangkan pada skenario kedua, secara default seluruh permintaan client dilayani oleh Cloudkilat sebagai primary server. Ketika terdapat kegagalan sistem dalam menjalankan layanannya seperti aplikasi dan database maka sistem akan secara otomatis mengalihkan layanan ke secondary server, dimana yang bertindak

yaitu server Google Cloud Engine. Dalam menjaga konsistensi data antara 2 cloud yang berbeda di setiap skenario terdapat proses replikasi database antara dua sistem.

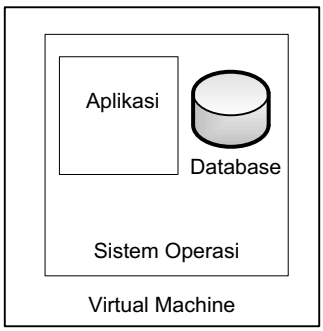
Pada tabel terdapat spesifikasi hardware dan software yang digunakan dalam penelitian ini

Tabel 3.1 Spesifikasi Hardware dan Software

Jenis	**CloudKilat**	**Amazon Web Service**	**Google Cloud Engine**
Hardware			
vCPU	2	1	1
Memory	2048 MB	1	4 GiB
Hardisk	40 GiB	30 GiB	30 GiB
Bandwitdh	Unlimited	Unlimited	Unlimited
Software			
Sistem Operasi	Ubuntu Server LTS 16.04	Ubuntu Server LTS 16.04	Ubuntu Server LTS 16.04
Web Server	Apache	Apache	Apache
Database Server	MySQL	MySQL	MySQL

3.1.2 Desain Aplikasi

Dalam pembangunan infrastruktur sistem Disaster Recovery as a Servise atau biasa disingkat DRaaS penentuan jenis dan tipe aplikasi merupakan salah satu hal yang terpenting dalam pengujian sistem. Aplikasi berbasis website dipilih sebagai aplikasi pengujian dikarenakan kemudahan dalam hal pembangunan dan penginstalan serta pemeliharaan dalam sistem[2].



Gambar 3.3 Desain Logic Aplikasi

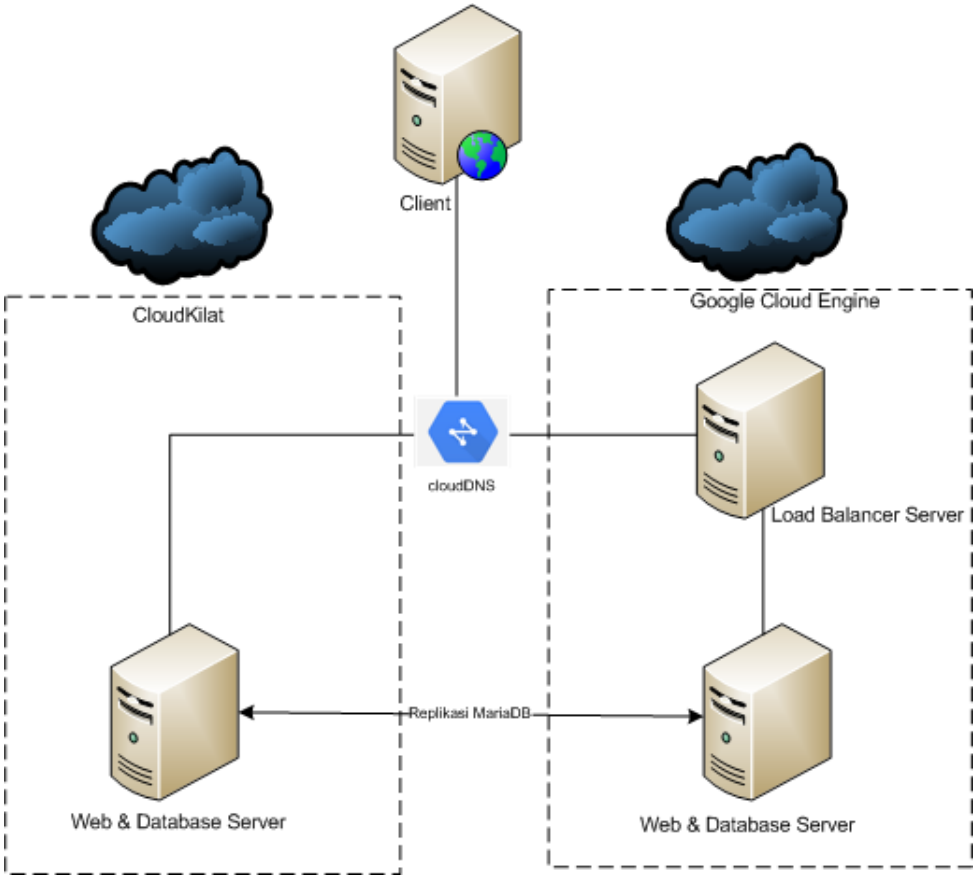
Pada Gambar 3.3 mengambarkan desain logic aplikasi dimana terdiri dari 4 komponen yaitu virtual machine, sistem operasi, aplikasi serta database. Virtual machine merupakan komponen terpenting dari cloud computing dimana menjadi inti dari layaan Disaster Recovery as a service. Tipe dan jenis dari virtual machine ini tentunya berbeda-beda tergantung dari masing-masing teknologi yang dipakai oleh tiap provider.

Komponen kedua yaitu sistem operasi merupakan layer 2 dari sistem dimana dalam penelitian ini menggunakan sistem operasi yang sama yaitu Ubuntu 16.04 LTS di virtual private server. Layer aplikasi merupakan layer ke 3 dimana layer ini yang berhadapan langsung dengan user yang melakukan pengujian layanan terhadap sistem yang dibangun. Aplikasi yang digunakan sebagai pengujian yaitu berbasis website dan berplatform e-commerce. Layer terakhir yaitu layer database merupakan layer terpenting dari sistem ini dimana menyimpan seluruh data aplikasi dari sistem, salah satu parameter pengujian yang akan dilakukan yaitu menjaga konsistensi database agar tetap konsisten diantara 2 server master dan slave

3.3 Implementasi Sistem

3.3.1 DRaaS Cloudkilat ke Google Compute Engine

Dalam skenario Disaster Recovery yang pertama dimana melibatkan dua provider cloud computing yaitu Cloudkilat dan Google Cloud Engine, infrastruktur disaster recovery yang dibangun dapat dilihat pada Gambar 3.4



Gambar 3.4 Skema Infrastruktur DRaaS Cloukilat ke GCE

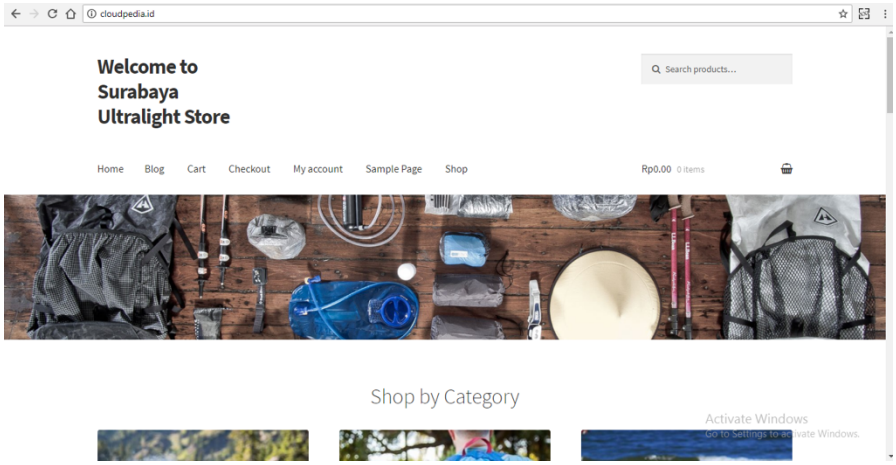
Pembangunan infrastruktur DRaaS dari provider Cloudkilat ke Google Compute Engine melalui beberapa tahapan, diantaranya

1. Pembangunan 2 server yaitu primary server dan secondary server yang terletak di dua provider. Primary server terletak di cloud Cloudkilat sedangkan secondary server terletak di Google Cloud Engine.
2. Pembangunan replikasi database diantara primary server dan secondary server yang didukung oleh teknologi replikasi database MySQL yang menjamin data diantara 2 server identik[17].
3. Pembangunan load balancer dimana menjamin proses failover/pengalihan layanan dari primary server dan secondary server yang didukung oleh layanan Google Cloud Engine yaitu Cloud DNS dan HA-Proxy

Tabel 3.2 Konfigurasi Infrastruktur Skenario Pertama

	Primary Server	**Secondary Server**	**Load Balancer Server**
Cloud Provider	Cloudkilat	Google Cloud Engine	Google Cloud Engine
IP Address Internal	-	10.140.0.2	10.160.0.2
IP Address Eksternal	103.43.45.223	35.201.210.241	35.200.145.200
Status Replikasi MariaDB	Master	Slave	-

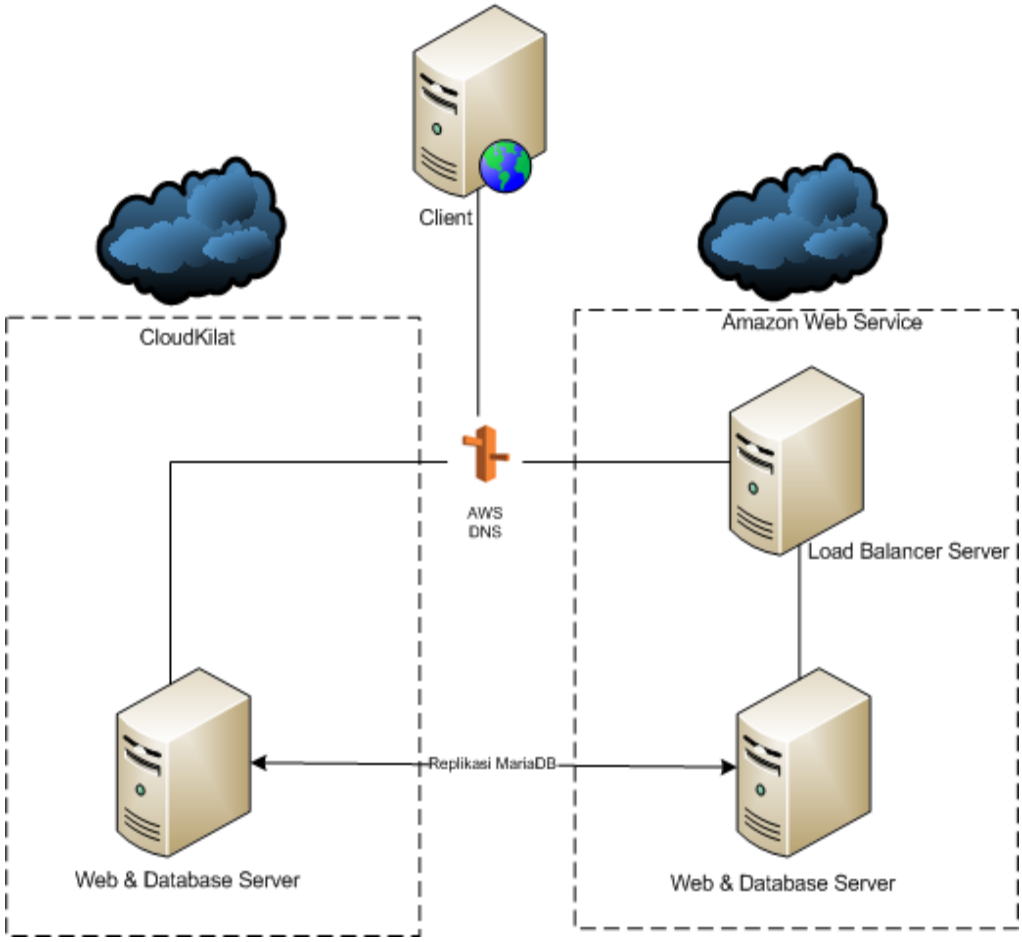
Pada Table 3.2 menjelaskan informasi konfigurasi yang dilakukan pada skenario pertama. Pada skenario pertama terdapat 3 VPC (Virtual Private Server) yang terlibat. 1 VPC yang bertindak sebagai primary server terletak di cloud provider Cloudkilat, 1 VPC yang bertindak sebagai secondary server yang berfungsi sebagai server yang mengambil alih layanan sistem ketika primary server sedang offline, dan yang terakhir yaitu 1 VPC sebagai load balancer yang berfungsi mengatur dan mengelola jaringan sistem antara primary dan secondary server.



Gambar 3.5 Tampilan layanan aplikasi

3.3.2 DRaaS Cloudkilat ke Amazon Web Service

Dalam skenario Disaster Recovery yang kedua dimana melibatkan dua provider cloud computing yaitu Cloudkilat dan Amazon Web Service, infrastruktur disaster recovery yang dibangun dapat dilihat pada Gambar 3.6



Gambar 3.6 Skema Infrastruktur DRaaS Cloukilat ke Amazon Web Service

Pembangunan infrastruktur DRaaS dari provider Cloudkilat ke Amazon Web Service melalui beberapa tahapan, diantaranya

1. Pembangunan 2 server yaitu primary server dan secondary server yang terletak di dua provider. Primary server terletak di cloud Cloudkilat sedangkan secondary server terletak di Amazon Web Service.
2. Pembangunan replikasi database diantara primary server dan secondary server yang didukung oleh teknologi replikasi database MySQL yang menjamin data diantara 2 server identik.

3. Pembangunan load balancer dimana menjamin proses failover/pengalihan layanan dari primary server dan secondary server yang didukung oleh HA-Proxy.

Tabel 3.3 Konfigurasi Infrastruktur Skenario Kedua

	Primary Server	**Secondary Server**	**Load Balancer Server**
Cloud Provider	Cloudkilat	Amazon Web Service	Amazon Web Service
IP Address Internal	-	10.140.0.2	10.160.0.2
IP Address Eksternal	103.23.22.76	18.220.46.86	18.219.189.47
Status Replikasi MySQL	Master	Slave	-

Pada Tabel 3.3 menjelaskan informasi konfigurasi yang dilakukan pada skenario pertama. Pada skenario pertama terdapat 3 VPC(Virtual Private Server) yang terlibat. Sebuah VPC yang bertindak sebagai primary server terletak di cloud provider Cloudkilat, sebuah VPC yang bertindak sebagai secondary server yang berfungsi sebagai server yang mengambil alih layanan sistem ketika primary server sedang offline, dan yang terakhir yaitu sebuah VPC sebagai load balancer yang berfungsi mengatur dan mengelola jaringan sistem antara primary dan secondary server.

Pada Gambar 3.5 menunjukkan tampilan aplikasi yang akan dilakukan uji coba pada pengujian skenario pertama. Domain yang bisa diakses untuk pengujian skenario ini yaitu http://ec2-18-219-189-47.us-east-2.compute.amazonaws.com. Pada kondisi normal user ketika mengakses cloudpedia.id akan diarahkan ke primary server yaitu pada Cloudkilat, sedangkan ketika primary server dalam

kondisi offline maka secara otomatis user akan diarahkan pada secondary server yaitu pada Amazon Web Server.

3.4 Kejadian Disaster

Untuk mengetahui apakah sistem disaster recovery yang telah dibangun dapat berjalan dengan baik maka perlu diuji dengan membuat sebuah kejadian disaster[7]. Disaster dalam penelitian ini disimulasikan dengan cara mematikan dengan sengaja virtual machine di dalam menu control panel cloud provider Cloudkilat yang bertindak sebagai primary server[13]. Kejadian disaster mengakibatkan layanan ecommerce menjadi menjadi offline atau tidak dapat diakses.

3.5 Skenario Pengujian dan Pengukuran Sistem

Pengukuran kinerja merupakan bagian dari sistem dimana mengevaluasi hasil implementasi sistem yang telah dilakukan. Dalam pengukuran kinerja terdapat beberapa metriks atau parameter yang digunakan yaitu :

1. Integritas Data dan RPO

   Digunakan sebagai metrik pengukuran kinerja Disaster Recovery yang bertujuan untuk memastikan konsistensi data antara 2 server tetap terjaga ketika terjadi event disaster[18]. Integritas dan konsistensi data berhubungan dengan metrik penting dalam Disaster Recovery yaitu RPO (Recovery Point Objective) merupakan satuan waktu dimana backup terakhir sistem sebelum terjadi *event disaster* dimana rentang waktu yang digunakan dalam penelitian ini yaitu 1-5 menit sesuai dengan range RPO yang didefinisikan oleh Suguna[16] .

   Pengujian integritas data dilakukan dengan cara :

   a. Menguji sistem backup file otomatis antara primary server dan secondary server di dua skenario yang berbeda. Pengujian dilakukan selama minimal 10 kali untuk memastikan bahwa data yang dibackup sesuai dan identik dengan data yang ada di primary server. Sistem

backup file terjadi secara otomatis per 3 menit sekali, sehingga pengecekan file harus dilakukan per 3 menit sekali.

b. Menguji integritas data yang langsung berhubungan dengan bisnis proses aplikasi yang digunakan user dengan cara menguji seluruh aktifitas dalam aplikasi website. Terdapat 8 aktifitas utama yaitu : Memesan Barang, Menambah Produk, Menghapus Produk, Menambah Review Produk, Menambah User Baru, Mengubah User, Menghapus User. Dari aktifitas utama diatas dibagi lagi menjadi 53 sub aktifitas.

   Pada pengujian ini, sebuah aktifitas akan diuji mulai dari sebelum terjadi failover, dan setelah terjadi failover untuk memastikan transaksi data pada setiap aktifitas tidak hilang ataupun corrupt setelah terjadi disaster, dan user dapat melanjutkan kembali aktifitas website yang terjadi tanpa harus mengulang dari awal.

2. Performansi

   Digunakan untuk membandingkan performansi server sebelum, saat, dan setelah event disaster dan failover. Komponen performansi yang digunakan yaitu CPU (Central Processing Unit) Utilization dan Memory Utilization. Server yang dijadikan objek monitoring yaitu secondary server di masing-masing infrastruktur yaitu di server Google Cloud dan Amazon Web Service.

   Pengujian performansi dilakukan dengan cara :

   a. Menyiapkan tools monitoring performansi pada server yang akan dilakukan pengujian yaitu secondary server.

   b. Pada tiap skenario pengujian infratsruktur dilakukan 8 aktifitas proses bisnis utama yang sama di waktu yang sama sehingga didapatkan perbedaan performansi secondary server dalam menangani aktifitas proses bisnis di skenario yang berbeda.

3. Waktu Recovery

Merupakan salah satu parameter penting dalam Disaster Recovery dikarenakan berhubungan langsung dengan aspek RTO (Recovery Time Objective) dimana mendefinisikan waktu yang dibutuhkan oleh sistem untuk dapat mengembalikan lagi akses aplikasi sistem yang sempat terhenti oleh gangguan bencana dalam satuan waktu[19].

RTO (Recovery Time Objective) merupakan satuan waktu dimana layanan aplikasi dapat diakses kembali setelah terjadi *event disaster*. Even disater yang dimaksud dalam pengujian ini diantaranya server master terjadi down atau offline, terjadi masalah pada koneksi jaringan, konfigurasi dsb yang menyebabkan layanan tidak dapat diakses.

Dalam penelitian ini nilai RTO didefinisikan dengan durasi maksium yaitu 60 menit berdasarkan range RTO yang didefinisikan oleh Suguna[16] .

Pengujian waktu recovery dilakukan dengan cara :

1. User melakukan pengaksesan aplikasi kemudian primary server dikondisikan dalam status offline. HA-Proxy yang bertindak sebagai load balancer akan mendeteksi status offline pada primary server dan akan mengalihkan pengaksesan aplikasi ke secondary server
2. Jeda waktu akan dihitung menggunakan aplikasi stop watch mulai dari saat aplikasi tidak dapat diakses karena primary server dalam kondisi off sampai HA-proxy mengalihkan pengaksesan aplikasi ke secondary server.
3. Pengujian akan dilakukan sampai dengan kesekian kali untuk mendapatkan nilai rata-rata dari tiap skenario, selanjutnya akan dibandingkan untuk menentukan skenario mana yang mempunya jeda waktu paling sedikit.

BAB 4
HASIL DAN PEMBAHASAN

4.1 Data Hasil Pengujian

4.1.1 Integritas Data

Integritas data merupakan salah satu aspek penting dalam pengujian Disaster Recovery[20]. Dalam penelitian ini, aplikasi website digunakan sebagai pengujian layanan dalam Disaster Recovery yang di bangun. Dalam pengelolaan website, data yang digunakan dibagi menjadi 2 komponen, yang pertama yaitu data file pendukung website seperti data multimedia seperti video, gambar, serta file pendukung lainnya dan yang kedua yaitu data dalam database penyimpanan.

1. Pengujian mirroring file data pendukung website

Pengujian mirroring data dilakukan untuk menguji integritas data website yang dibackup dari primary server ke secondary server. Sistem backup file yang telah berlangsung berjalan setiap 3 menit sekali oleh sistem.

Tabel 4.1 Pengujian backup file

No	Mirroring Data Time (Minutes)	CloudKilat to Google Cloud	CloudKilat to Amazon Web Service
1	0 – 3	Succeeded	Succeeded
2	3 – 6	Succeeded	Succeeded
3	6 – 9	Succeeded	Succeeded
4	9 – 12	Succeeded	Succeeded
5	12 – 15	Succeeded	Succeeded
6	15 – 18	Succeeded	Succeeded
7	18 – 21	Succeeded	Succeeded
8	21 – 24	Succeeded	Succeeded
9	24 – 27	Succeeded	Succeeded
10	27 – 30	Succeeded	Succeeded

Dari data hasil pengujian pada Tabel 4.1 didapatkan bahwa pada kedua jenis skenario pengujian yaitu dari Cloudkilat ke Google Cloud dan Cloudkilat ke Amazon Web Service data backup pada secondary server selalu konsisten mengikuti data file dari primary server. Data website yang berhasil dibackup setiap 3 menit menjadi nilai RPO untuk masing-masing skenario.

2. Pengujian Konsistensi data pada aplikasi

Pengujian konsistensi data pada aplikasi bertujuan untuk memastikan layanan aplikasi ketika terjadi failover dapat menjaga data dari aktifitas aplikasi sebelum terjadi failover, sehingga ketika layanan dapat diakses kembali user dapat melanjutkan aktifitas yang sempat terhenti akibat terjadi failover. Data yang berperan dalam pengujian ini yaitu data pada database. Hasil skenario 1 merupakan pengujian dari Cloudkilat ke Google Cloud dan hasil skenario 2 merupakan hasil pengujian dari CloudKilat ke Amazon Web Service.

Tabel 4.2 Pengujian konsistensi data pada aplikasi

No	Aktifitas	Detail Aktifitas	Aktor	Lokasi	Hasil Skenario 1	Hasil Skenario 2
1	Memesan Barang	1.1 Mengakses Halaman Homepage	Customer	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		1.2 Memilih Produk	Customer	Primary Server		
		1.3 Masuk ke Keranjang Belanja	Customer	Primary Server		
		1.4 Mengubah Keranjang Belanja	Customer	Primary Server		
		1.5 Melakukan Proses Checkout	Customer	Primary Server		
		1.6 Mengisi Form Registrasi Akun	Customer	Primary Server		
		1.7 Mengisi Form Alamat Pengiriman	Customer	Primary Server		
		1.8 Memilih Metode Pembayaran	Customer	Primary Server		

		1.9 Memesan Barang	Customer	Primary Server		
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		1.10 Memproses Pemesanan	Administrator	Secondary Server		
		1.11 Melihat Status Pemesanan	Customer	Secondary Server		
2	Menambah Produk	2.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		2.2 Mengakses Halaman Produk	Administrator	Primary Server		
		2.3 Mengakses Halaman Add Product	Administrator	Primary Server		
		2.4 Mengisi Deskripsi Produk	Administrator	Primary Server		
		2.5 Mengunggah Gambar Produk	Administrator	Primary Server		
		2.6 Mempublish Produk	Administrator	Primary Server		
		2.7 Menampilkan Produk	Sistem	Primary Server		
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		2.8 Melihat Produk Baru	Administrator	Secondary Server		
		2.9 Melihat Produk Baru	Customer	Secondary Server		
3	Mengubah Produk	3.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		3.2 Mengakses Halaman Produk	Administrator	Primary Server		
		3.3 Mengakses Halaman Edit Produk	Administrator	Primary Server		
		3.4 Mengubah Informasi Produk	Administrator	Primary Server		
		3.5 Mengupdate Informasi Produk Terbaru	Sistem	Primary Server		

		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		3.6 Melihat Produk	Administrator	Secondary Server		
		3.7 Melihat Produk	Customer	Secondary Server		
4	Menghapus Produk	4.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		4.2 Mengakses Halaman Produk	Administrator	Primary Server		
		4.3 Memindah Produk Ke Menu Trash	Administrator	Primary Server		
		4.4 Mengakses Halaman Trash	Administrator	Primary Server		
		4.5 Menghapus Permanen Produk	Administrator	Primary Server		
		4.6 Menghapus Permanen Produk	Administrator	Primary Server		
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		4.7 Data Terhapus	Administrator Customer	Secondary Server		
5	Menambah Review Produk	5.1 Mengakses Homepage	Customer	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		5.2 Mengakses Detail Produk	Customer	Primary Server		
		5.3 Login	Customer	Primary Server		
		5.4 Mengisi Kolom Review	Customer	Primary Server		
		5.5 Submit Review	Customer	Primary Server		
		5.6 Menyetujui Review	Administrator	Primary Server		
		5.7 Menampilkan Review Customer	Sistem	Primary Server		

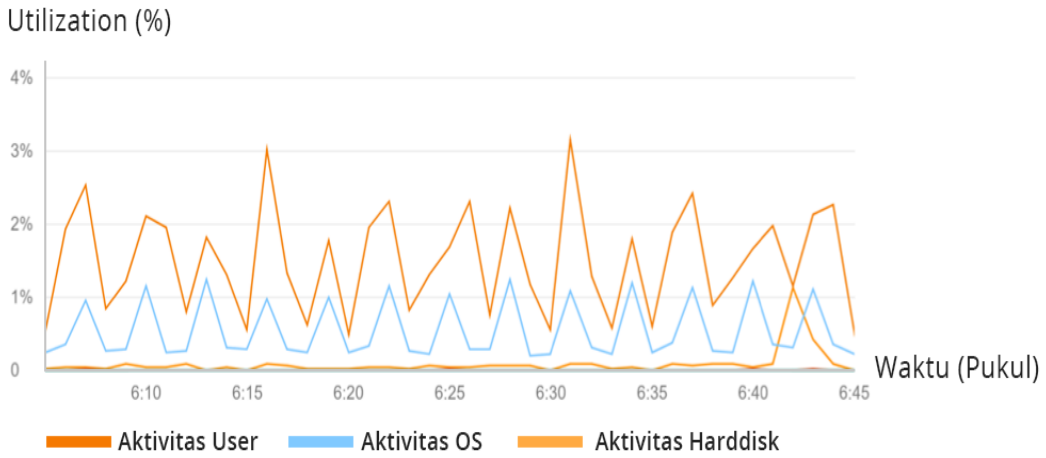
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		5.8 Melihat Review Customer	Administrator	Secondary Server		
		5.9 Melihat Review	Customer	Secondary Server		
6	Menambah User Baru	6.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		6.2 Mengakses Halaman User	Administrator	Primary Server		
		6.3 Mengakses Halaman Tambah User	Administrator	Primary Server		
		6.4 Mengisi Informasi User Baru	Administrator	Primary Server		
		6.5 Menyimpan Data User Baru	Administrator	Primary Server		
		6.6 Menampilkan Data User Baru	Sistem	Primary Server		
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		6.7 User baru login	Customer/ Administrator	Secondary Server		
7	Mengubah User	7.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		7.2 Mengakses Halaman User	Administrator	Primary Server		
		7.3 Mengakses Halaman Ubah User	Administrator	Primary Server		
		7.4 Mengubah Data User	Administrator	Primary Server		
		7.5 Menyimpan Perubahan User	Administrator	Primary Server		
		Failover (Terjadi kondisi offline pada primary server dan secondary server mengambil alih pelayanan aplikasi)				
		7.6 Menampilkan Data User Terbaru	Sistem	Secondary Server		

8	Menghapus User	8.1 Login	Administrator	Primary Server	Berhasil, data tetap konsisten setelah terjadi failover	Berhasil, data tetap konsisten setelah terjadi failover
		8.2 Mengakses Halaman User	Administrator	Primary Server		
		8.3 Memilih Menu Hapus User	Administrator	Primary Server		
		8.4 Konfirmasi Penghapusan User	Administrator	Primary Server		
		8.5 Menghapus Data User	Sistem	Primary Server		
		8.6 Data terhapus	Sistem	Secondary Server		

4.1.2 Pengujian Performansi

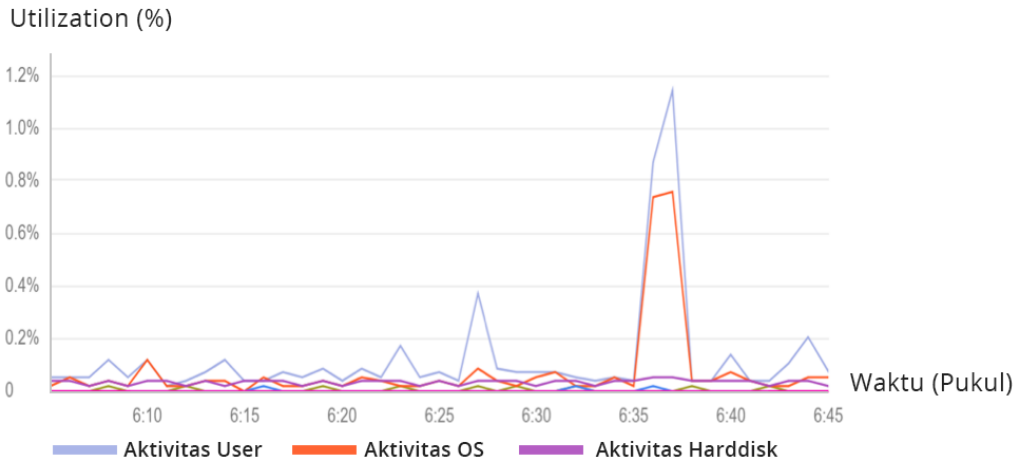
Pengujian perfomansi yang dilakukan dalam penelitian ini bertujuan untuk mengatahui behaviour dan perbandingan performansi antara skenario 1 dan skenario 2. Parameter performansi yang diuji yaitu CPU utilizations dan Memory utilization, sedangkan untuk server yang dimonitor yaitu secondary server pada tiap skenario. Pengujian dilakukan dengan melakukan lagkah-langkah yang sama yaitu 8 aktifitas utama pada aplikasi website di tiap skenario dan dilakukan selama 40 menit yaitu dari jam 06.05 sampai dengan 06.45.

1) Perbandingan CPU Utilization



Gambar 4.1 CPU Utilization GCE

Pada Gambar 4.1 menunjukkan grafik CPU Utilization pada secondary server di server Google Cloud. Sumbu x merupakan waktu pengaksesan sedangkan sumbu y merupakan nilai presentasi dari CPU utilizations. Sumbu kuning mempresentasikan parameter ‘user’ dari CPU yang berarti presentase pemakaian CPU untuk layer paling tinggi dalam sistem operasi yaitu aplikasi termasuk website dan database. Pada grafik terlihat bahwa presentase CPU tertinggi berada pada titik 3,15% dan terendah pada 0, 47%.



Gambar 4.2 CPU Utilization AWS

Pada Gambar 4.2 menunjukkan grafik CPU Utilization pada secondary server di server Amazon Web Service. Sumbu x merupakan waktu pengaksesan sedangkan sumbu y merupakan nilai presentasi dari CPU utilizations. Sumbu biru mempresentasikan parameter ‘user’ dari CPU yang berarti presentase pemakaian CPU untuk layer paling tinggi dalam sistem operasi yaitu aplikasi termasuk website dan database. Pada grafik terlihat bahwa presentase CPU tertinggi berada pada titik 1.14% dan terendah pada 0, 2%.

2) Perbandingan Memory Utilization

Pada Gambar 4.3 menunjukkan grafik Memory Utilization pada secondary server di server Google Cloud. Sumbu x merupakan waktu pengaksesan sedangkan sumbu y merupakan nilai presentasi dari memory utilizations. Sumbu merah pada grafik menunjukkan parameter ‘used’ yang berarti presentase pemakaian memory

pada secondary server. Pada grafik terlihat bahwa presentase memory tertinggi berada pada titik 10.971% dan terendah pada 10,686%

Gambar 4.3 Memory Utilization GCE

Gambar 4.4 Memory Utilization AWS

Pada Gambar 4.4 menunjukkan grafik Memory Utilization pada secondary server di server Amazon Web Services. Sumbu x merupakan waktu pengaksesan sedangkan sumbu y merupakan nilai presentasi dari memory utilizations. Sumbu biru pada grafik menunjukkan parameter ‘used’ yang berarti presentase pemakaian

memory pada secondary server. Pada grafik terlihat bahwa presentase memory tertinggi berada pada titik 51,85% dan terendah pada 51,690%.

4.1.3 Pengujian Waktu Recovery Aplikasi

Pengujian jeda aplikasi dilakukan untuk menghitung jeda dalam detik pengaksesan layanan aplikasi ketika proses failover (proses pengalihan layanan dari primary server ke secondary server ketika disaster terjadi) .

1. Cloudkilat ke Google Cloud

Pengujian jeda aplikasi skenario CloudKilat ke Google Cloud dilakukan dengan cara menghentikan layanan aplikasi pada server Cloudkilat dengan membuat server Cloudkilat berstatus offline, kemudian dihitung seberapa cepat user dapat mengakses kembali website cloudnesia.id saat layanan telah diambil oleh server Google Cloud.

Tabel 4.3 Tabel Pengujian Waktu Recovery Aplikasi Skenario 1

Pengujian ke-	Jeda (detik)
1	13
2	14
3	11
4	14
5	17
6	20
7	18
8	21
9	14
10	12
Rata-Rata	15,4

Pada Tabel 4.4 terdapat hasil pengujian jeda aplikasi setelah proses failover yang dilakukan selama 10 kali dalam waktu yang berbeda. Dari hasil pengujian yang dihasilkan rata-rata jeda layanan aplikasi pada skenario 1 yaitu 15,4 detik.

2. Cloudkilat ke Amazon Web Service

Pengujian jeda aplikasi skenario CloudKilat ke Amazon Web Service dilakukan dengan cara menghentikan layanan aplikasi pada server CloudKilat dengan membuat server Cloudkilat berstatus offline, kemudian dihitung seberapa cepat user dapat mengakses kembali website http://ec2-18-219-189-47.us-east-2.compute.amazonaws.com saat layanan telah diambil oleh server Amazon Web Service.

Tabel 4.4 Tabel Pengujian Waktu Recovery Aplikasi Skenario 2

Pengujian ke-	Jeda (detik)
1	12
2	11
3	9
4	9
5	7
6	10
7	11
8	12
9	9
10	6
Rata-Rata	9.6

Pada Tabel 4.5 terdapat hasil pengujian jeda aplikasi setelah proses failover yang dilakukan selama 10 kali dalam waktu yang berbeda. Dari hasil pengujian yang dihasilkan rata-rata jeda layanan aplikasi pada skenario 2 yaitu 9,6 detik.

4.2 Analisis hasil Pengujian

Pada bagian ini akan dibahas analisis data hasil pengujian yang didapat dari bab 4.1

4.2.1 Analisis Pengujian Integritas Data

Pengujian integritas data dilakukan untuk memastikan bahwa data yang tersimpan terjamin konsistensinya saat terjadi kejadian disaster atau bencana.

Dalam sistem Disaster Recovery integritas dan konsistensi data berhubungan langsung dengan salah satu parameter terpenting dalam sistem DRaaS yaitu RPO (Recovery Point Objective). Ketika user sedang melakukan aktifitas tertentu pada aplikasi sistem dan tiba-tiba terjadi offline pada primary sistem, dan secondary sistem mengambil alih dalam pelayanan aplikasi, data user dapat terjamin dan dapat melanjutkan aktifitas yang terhenti sementara tanpa mengkhawatirkan data yang masuk tepat sebelum kejadian offline server. Dalam sistem Disaster Recovery as a Service yang dibangun dalam penelitian ini menggunakan 2 teknik untuk memastikan integritas data terjamin yaitu,

1. Backup file secara periodik dari primary server ke secondary server yang dilakukan secara otomatis per 3 menit.
2. Replikasi database MariaDB berbasis MySQL dengan mode master-slave dari primary server ke secondary server.

Dari hasil pengujian yang didapatkan pada tabel 4.1, didapatkan hasil bahwa pada 2 skenario yang diuji yaitu CloudKilat ke Google Cloud, dan CloudKilat ke Amazon Web Service, sama-sama berhasil dalam proses backup file server secara periodik selama 3 menit yang selama pengujian dipantau selama 30 menit. Durasi 3 menit back up file menjadi nilai RPO yang dihasilkan oleh masing-masing skenario. Hal ini memenuhi kriteria RPO yang telah didefinisikan dalam penelitian ini yaitu dalam rentang 0-5 menit berdasarkan tabel pemetaan yang dibuat oleh Suguna[16] dimana penelitian ini termasuk dalam kriteria tier 7 yaitu sistem dengan "Mirrored data with failover".

Selain menguji file backup secara periodik, pengujian langsung terhadap aktifitas user yang sempat terhenti karena adanya offline pada primary server juga penting dilakukan agar user dapat melanjutkan kembali aktifitas yang sempat terhenti karena kejadian disaster yang menyebabkan primary server tidak dapat diakses. Dari pengujian pada tabel 4.2 mendapatkan hasil bahwa dari sejumlah aktifitas proses bisnis yang dilakukan pada website yaitu sebanyak 8 aktifitas inti yaitu, Memesan Barang, Menambah Produk, Menghapus Produk, Menambah Review Produk, Menambah User Baru, Mengubah User, Menghapus User, sistem

dapat menyimpan data terakhir pada sebuah aktifitas transaksi sebelum terjadi offline pada primary server, dan user dapat melanjutkan aktifitas kembali dengan akses layanan aplikasi ke secondary server.

4.2.2 Analisis Pengujian Performansi

Pada tabel 4.5 menyajikan hasil pengujian yang didapatkan dari monitoring performansi secondary server dari setiap skenario pengujian.

Tabel 4.5 Perbandingan Hasil Pegujian Performansi

No	Komponen Pengujian	Hasil Skenario 1(Cloudkilat ke Google Cloud)	Hasil Skenario 2(Cloudkilat ke Amazon Web Service)
1	CPU Utilization Tertinggi	3,15 %	1,14%
2	CPU Utilization Terendah	0, 47%.	0,42%
3	Memory Utilization Tertinggi	10.971%	51,85%
4	Memory Utilization Terendah	10,686%	51,690%

Berdasarkan hasil data yang didapatkan server Google Cloud lebih unggul pada komponen Memory, sedangkan Amazon Web Service lebih unggul dari segi CPU. Pada perbandingan memory Google Cloud lebih unggul dibanding Amazon Web Service dikarenakan spesifikasi memory yang digunakan lebih tinggi dari Amazon Web Service. Server Google Cloud menggunakan memory sebesar 4 GiByte sedangkan Amazon Web Service hanya menggunakan memory sebesar 1 GiByte. Untuk spesifikasi CPU, kedua server memiliki nilai yang sama yaitu hanya mempunyai 1 Virtual CPU. Sehingga dapat disimpulkan untuk pengujian performasi skenario 2 yang lebih unggul dibanding skenario 1.

4.2.3 Analisis Pengujian Waktu Recovery Aplikasi

Pengujian waktu recovery aplikasi bertujuan untuk mengetahui seberapa lama waktu aplikasi dapat diakses kembali setelah terjadi bencana/disaster. Pengujian jeda aplikasi sangat erat kaitannya ketika dihubungkan dengan parameter terpenting dalam Disaster Recovery yaitu RTO (Recovery Time Objective). Pada tabel 4.4 dan tabel 4.5 didapatkan hasil dari masing-masing skenario. Rata-rata jeda pada skenario pertama didapatkan nilai 15, 4 detik, sedangkan untuk skenario kedua didapatkan nilai rata-rata 9,6 detik. Oleh karena itu, skenario kedua dalam hal ini AWS lebih unggul karena memiki nilai RTO lebih cepat sebesar 37.66% dibanding Google Cloud. Perbedaan waktu recovery antara 2 skenario ini dapat disebabkan oleh berbagai kemungkinan diantaranya yaitu :

1. Jarak antara primary server dan secondary server

   VPS CloudKilat terletak di Indonesia, sedangkan dua provider lainnya yaitu Google Cloud dan Amazon Web Service tidak terletak di Indonesia yaitu di Amerika Serikat. Untuk Google Cloud, server terletak di South Carolina, sedangkan VPS di Amazon Web Service terletak pada Ohio.

2. Bandwidth yang digunakan

   Bandwidth merupakan salah satu komponen yang dapat mempengaruhi jeda aplikasi dimana alokasi bandwidth yang berbeda-beda di tiap provider.

3. Padatnya jaringan yang digunakan antara secondary dan primary server juga dapat mempengaruhi perbedaan jeda aplikasi.

Hal ini memenuhi kriteria RTO yang telah didefinisikan dalam penelitian ini yaitu dalam rentang 0-5 menit berdasarkan tabel pemetaan yang dibuat oleh Suguna[16] dimana penelitian ini termasuk dalam kriteria tier 7 yaitu sistem dengan “Mirrored data with failover”.

Halaman ini sengaja dikosongkan

BAB 5
KESIMPULAN DAN SARAN

Pada bab ini akan diuraikan beberapa kesimpulan dan saran yang dapat diambil dari pembahasan sebelumnya dan saran mengenai masalah yang bisa dibahas sebagai kelanjutan dari penelitian ini.

5.1 Kesimpulan

Kesimpulan yang didapatkan dari pengujian yang sudah dilakukan adalah sebagai berikut:

1. Pembangunan infrastruktur Disaster Recovery as a Service dengan 2 skenario berbeda yaitu pertama dari CloudKilat ke Google Cloud Engine dan kedua dari Cloudkilat ke Amazon Web Service telah berhasil dilakukan
2. Berdasarkan hasil pengujian fungsionalitas layanan aplikasi setelah terjadi failover, aplikasi tetap dapat diakses melalui secondary server meskipun terjadi offline pada primary server.
3. Berdasarkan hasil pengujian integritas data setelah terjadi failover, konsistensi data tetap terjamin walaupun primary server dalam kondisi offline.
4. Berdasarkan hasil pengujian, kedua skenario telah berhasil memenuhi nilai RTO maksimum yang telah ditentukan yaitu 60 menit. Skenario kedua dimana Amazon Web Service sebagai secondary server menunjukkan waktu recovery rata-rata lebih cepat yaitu 9.6 detik dibanding skenario pertama yaitu 15,4 detik.
5. Kinerja sistem Disaster Recovery as a Service menggunakan Amazon Web Service lebih unggul dibanding Google Cloud berdasarkan perbadingan parameter RTO
6. Berdasarkan hasil pengujian, kinerja kedua skenario baik Cloudkilat dengan Google Cloud dan CloudKilat dengan AWS adalah seimbang karena menghasilkan nilai RPO yang seimbang yaitu sebesar 3 menit dan telah memenuhi ketentuan sistem disaster recovery menggunakan metode mirrored data with failover dengan nilai RPO di bawah 5 menit.

5.2 Saran

Berdasarkan hasil perancangan sistem dan pengujian yang telah dilakukan, dapat diberikan beberapa saran untuk pengembangan *Disaster Recovery as a Service* adalah sebagai berikut:

1. Menggunakan cloud provider lain di berbagai lokasi data center agar mendapatkan hasil pengujian terkait pengaruh jarak.
2. Menambahkan variasi jumlah user yang mengakses aplikasi atau layanan untuk mendapatkan hasil RTO rata-rata yang dialami oleh semua user.

DAFTAR PUSTAKA

[1] P. Suraj, M. Sneha, W. Abdul, and S. Sundaram,"Disaster recovery services in the cloud for SMEs," International Conference on Cloud ComputingTechnologies, Applications and Management (ICCCTAM) , pp.139-144, 8-10 Dec. 2012.

[2] A. Mohammad Matar, A. Ali A, and A. Imad Fakhri, "Data recovery and business continuity in Cloud computing: A Review of the Research Literature", International Journal of Advancements in Computing technology, December 2016

[3] Google Cloud Solution, "Disaster Recovery Cookbook", Available https://cloud.google.com/solutions/disaster-recovery-cookbook.

[4] A. Omar H., M. Yashwant .K, "Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud ", 2012 IEEE 23rd International Symposium on, Dallas, TX, 27-30, Page(s): 19 – 20, November 2012

[5] Y. Shindura, "Effectiveness of Backup and Disaster Recovery in Cloud : Comparative study on Disk and Cloud based Backup and Disaster Recovery". Karlskrona: Blekinge Institute of Technology.

[6] R. Glen, N. Attila, and E. Chris,"Using Amazon Web Services for Disaster Recovery", Available: https://media.amazonwebservices.com/AWS_Disaster_Recovery.pdf

[7] G. Peter, "IT Disaster Recovery For Dummies", Indiana : Wiley Publishing,Inc. 2008.

[8] P. Manish, L. Seulki, and P. Jong Sou, "Disaster Recovery for System Architecture using Cloud Computing", 10th IEEE/IPSJ International Symposium on Seoul,Page(s):304–307, 19-23 July 2010

[9] Barry Sosinky, "Cloud Computing Bible", Indianapolis : Wiley Publishing,Inc. 2011

[10] A. Dario, C. Manuel, H. Ileana, and M. Sergio, “The Use of Cloud Computing in SMEs”, Elsevier International Journal Mathematical and Computer Modelling, Procedia Computer Science 83 (2016) 1207 – 1212, 2016

[11] Amazon Web Service, “Amazon Web Service” Available https://aws.amazon.com/

[12] Google Cloud, “Why Google Cloud ”, Availlable https://cloud.google.com/why-google-cloud/

[13] Cloudkilat, “F&Q about Cloukilat”, Available http://www.cloudkilat.com/

[14] D. Alex., “High Availability MySQL Cookbook”, Packt Publishing Ltd., Birmingham, 2010

[15] M. Hossam Abdel Rahman, “A Proposed Model for IT Disaster Recovery Plan”, I.J. Modern Education and Computer Science, pp57-67, April 2014

[16] S. Suguna and A. Suhasini, “Overview of data backup and disaster recovery in cloud”. International Conference on Information Communication and Embedded Systems (ICICES2014) (978) 1–7, February 2014

[17] Oracle Corporation, “MySQL 5.5 Manual Replication”, Available, http://dev.mysql.com/doc/refman/5.5/en/replication.html

[18] P.Athanasios, and Z. Tomas, “Criticality estimation of IT business functions with the Business Continuity Testing Points method for implementing effective recovery exercises of crisis scenarios”, International Journal of Computer Science Issues, November 2013.

[19] S. Kruti and S. Kavita R, “Online Data Back-up and Disaster Recovery Techniques in Cloud Computing”, International Journal of Engineering and Innovative Technology (IJEIT) Vol 2 Issue 5, pp249-254, November 2012.

[20] G. Peter , “DRaaS For Dummies® Veeam® Software Special Edition”, New Jersey:John Wiley & Sons, 2016.

LAMPIRAN

Paper yang dipublikasikan pada konferensi ICITEE 2018

Analysis of RTO and RPO of a Service Stored on Amazon Web Service (AWS) and Google Cloud Engine (GCE)

Yorisan P. Baginda
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
yorisan@elect-eng.its.ac.id

Achmad Affandi
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
affandi@ee.its.ac.id

Istas Pratomo
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
istaspra@ee.its.ac.id

***Abstract*— Today, the availability of the application is beyond everything. Application can not be accessed even in a minute will become major problem for the business and threat the reputation of the company especially that need 24/7 availability of application. Companies that have traditional physical environments typically must duplicate their infrastructure to ensure the availability of spare capacity in the event of a disaster. Nowadays, there are many cloud providers that offer service called DRaaS (Disaster Recovery as a Service) to facilitate company needs in case of Disaster Recovery for their system to ensure their business continuity. Many companies still confuse which provider that can suit their system. Since RTO and RPO is the most critical two metrics in disaster recovery planning. This paper introduces implementation design method and analysis of two metrics between two cloud providers.**

Keywords—DRaaS; AWS; Google Cloud;

I. INTRODUCTION

The emergence of cloud computing technology has changed the paradigm of the business world in applying technology in its business processes massively. The earlier cloud services based on pay-per-use models, IaaS(Infrastructure as a Service), PaaS(Platform as a Service), and SaaS(Software as a Service) have made many businesses previously using traditional infrastructure system moved to cloud-based infrastructure to improve the reliability of their business based. In their system usually include many applications that become core business of the company.

Today, the availaibility of the application is beyond everything. Application can not be accessed even in a minute will become major problem for the business and threat the reputation of the company especially that need 24/7 availability of application. Disaster is one of many reason that system or application can be unavailable to deliver service. Fires, water main breaks, hurricanes, floods, telecom outages, even hardware failures, software bugs, administrator error results in data corruption or deleted objects could be happened considered as disaster could happen in daily basis [1]. If company's system including all applications and database are affected and provoke downtime, this may be the substansial problem even downfall for the business.

Disaster recovery planning came up as one of many attempts that can save our system from undesirable disaster. Amazon Web Service (AWS) [2] define disaster recovery is anything in respect of preparing for and recovering from a disaster. Everything that has bad effects on a company's business sustainability could be termed a disaster. This belonging hardware or software failure, a network outage, a power outage, physical damage to a building like fire or flooding, human error, or some other significant event[2].

RTO(Recovery Time Objective) and RPO(Recovery Point Objective) is the most critical two metrics in disaster recovery planning[3]. RTO value means the duration of service unavailable until recover and start the service again. In the other hand, RPO means maximum amount of data that can be lost when restoration is successful in time. A successful disaster recovery key consists of maintain the service always online according to SLA also meeting RPO and PTO goals[4].

Cloud computing establish high reliability of service for preserving sensitive and important client data[5]. According to Assante et al [6] utilizing latest trend of technology that is cloud computing for companies is one of strategic skills to maintain the companies existance between competitors.

Today, there are lot of DRaaS provider such as AWS, Google Cloud, OVH that offered disaster recovery in their service. Every provider almost offered same features that confusing customers. Therefore we conduct several testing between providers that focus on RTO and RPO values to help customer for seeking the best choice for their disaster recovery planning.

II. BACKGROUND

Disaster Recovery is an effort to preparing and recovering from a diseaster. Everything that has bad effects on a company's business sustainability could be termed a disaster. This belonging hardware or software failure, a network outage, a power outage, physical damage to a building like fire or flooding, human error, or some other significant event. Every company must have its own infrastructure to support the business process especially that needs strong demand of application availability. Therefore, disaster recovery plan is not an option but it is a must.

Many advantages could be achieve applying disaster recovery plan, in-between assist minimize defect of critical functions, recover operations quickly and successfully when system is down, assure stabilityof organization, also preserves organization's assets[7]. Moreover, disaster recovery plan permit organization or company to state higher availability and reliability of services than other competitors[8].

There are three approaches to build disaster recovery, hot site, cold site, and warm site. Hot site means that companies have exact duplicate of their entire architecture system including server, data, storage, and network called secondary location that will not be affected by any event affected the primary location. The data is replicated between primary and secondary site. It could be active-active replication, where all databases are sync each other and replicated in two-way direction or active-passive replication, when data is replicated in one way direction. Meanwhile, Cold site is the type of approach that just send backup file of the system. Warm site denote average way between cold and hot.

In disaster recovery terms, the mechanism to switch service from primary site to secondary site when the service failed due to disaster is called failover[9]. In failover state, all services including application and data will be handled by secondary site. Otherwise, when the primary site problem has been resolved and the service is switch back to the primary site its called failback.

Today, building a disaster recovery system in traditional architecture has so much disadvantages such as complexity and cost including operating, maintaining, human resource and building cost[10]. Diseaster Recovery as a Service emerge to overcome cost and complexity problem in physical infrastructure.

Disaster Recovery as a Service is cloud based service responsible to ensure system availability and recovery from disaster. DRaaS brought different way to back up substantial system including application, data also resources and quickly recover systems after a disaster with less cost and complexity. The advantages of implementing DRaaS are Availability, Cost Reduction, Simplicity, Visibility, Scalability, Flexibility. Moreover we are allow to conducting DR testing in case the disaster is really happen as a trial.

In order to achive well-prepared disaster recovery plan, several metrics must be set. Podaras et al[11] explain various parameter in recovery test i.e Critical Business Function (CBF), Maximum Acceptable Outage (MAO); Recovery Time Objectives (RTOs), and Business Impact Analysis (BIA). In the other hand, Wood et al[12] specify other metrics such as RPO, Performance, Consistency, Geographic Separation.

In this research we will use two metrics to analysis our recovery disaster testing between providers. RTO stands for Recovery Time Objectives means the duration of service for example an application unavailable until recover and start the application service again. The less time required, the faster the recoveries will need to be.

In the other hand, RPO means maximum amount of data that can be lost when restoration is successful in time. RPO declares the data might be lost during disaster, the lower the RPO, the higher the total cost to save primary infrastructure environment for recovery [12]. In order to decrease the amount of data lost, perform a frequent backup scheduling might be the best choice. Figure 1 depicts a very clear distinction between RTO and RPO.

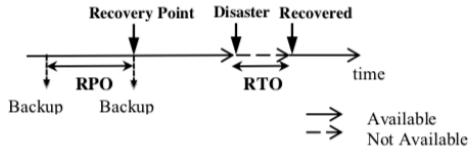


Fig. 1. The distinguish between RTO and RPO[12]

Amazon Web Service and Google Cloud Engine are two of hundreds provider serve disaster recovery as a service in one of their products. AWS(Amazon Web Service) offered services and features to meet customer requirement in their disaster recovery plan including storage, server, networking, database, deployment orchestration equipped with interactive tools[13]. Since AWS have multiple region site around the word, customer could choose the most appropriate location site.

AWS have several approaches to implement disaster recovery strategy. There are 4 possible approaches i.e Backup & Restore, Pilot Light, Warm Standby, and multisite. Every approach has own advantages and disadvantages regarding customer business process needs. Backup and Restore option allowed us to backup data in our system and sent to Amazon S3, cloud storage provided by Amazon Web Service, through the network. Pilot light option is similar to previous backup and restore option, with minimal version of environment running on the cloud. Pilot light uses data mirroring replication to ensure the consistency of data.

Warm standby is extend infrastructure of pilot light version. We have duplicate system on AWS that is always running thus decreases recovery time. Load balancer play a role as failover mechanism between on-premise and AWS. The last one is multi site option that offered smallest RTO and RPO values that is real time failover system on the cloud. This option supported by active-active replication of database.

Google Compute Engine also have solutions model for Disaster Recovery as a Service especially for application

recovery. Hot standby, Warm standby and Cold standby are solution offered for applications running on Google Cloud Engine[13].

III. PROPOSED METHOD

The method used in this research in adapting Hossam's data centre recovery phase. The recovery phase is divided in 3 sub-phase. The first phase is disaster recovery assessment plan. The next phase is disaster recovery action, and the last is system and application testing, also the analysis after testing.

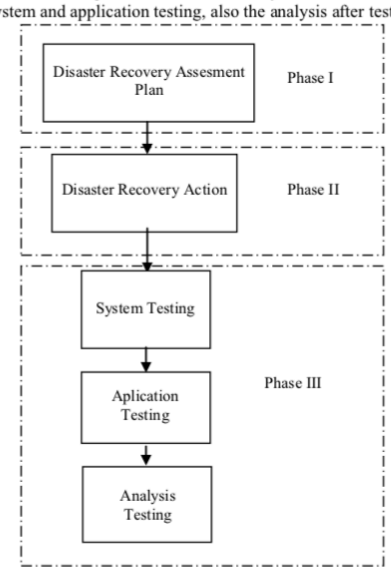


Fig. 2. Implementation Phases

Figure 2 explain the disaster recovery plan of this research. Disaster recovery assessment plan contains all activities about design system and application will be testing in this research, also design procedures for entire steps. The main metrics for evaluating diseaster recovery will be set and define in this phase along with testing and evaluating procedures scenario.

The second phase, disaster recovery action is the phase where all of the plans and designs will be executing. Deploying secondary system architecture, verify the operating systems and communication software between two sites are working properly also implement backup procedures and failover method will be done in this phase.

System and application testing helps to verify that the recovery procedures work as expected. Several testing will be repeated to ensure reliability of the system. Analysis phase will be held when all procedures testing are done and will be release suggestion for improvement when necessary.

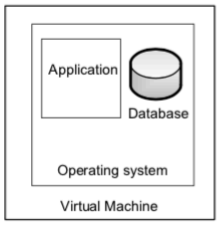


Fig. 3. System application logical tier

Figure 3 explain application testing environment used in this research. Application run inside operating system that lied in virtual machine environment since the primary server infrastructure is located in the cloud dedicated server. Since the the main concern of diseaster recovery is preserving critical data and system especially application that run core business of the company. We use database replication to maintain consistency between primary and secondary system. We will use Ubuntu 16.04 LTS as operating system, web-based application, and MySQL as database server in three different cloud systems. However virtual machine type might vary in each cloud system since we will use three different cloud providers as implementation testing.

Testing scenario in this disaster recovery implementation is divided in 2 scenario. The first testing scenario is between CloudKilat as primary server then Amazon Web Service as secondary server. The consistency data between two system was guaranteed by master-slave replication. As normal activity user will be directed to primary server, when disaster is occured system will be automatically switched to secondary server. This mecanism called failover that ensure the availability of system, therefore user client still could access the service

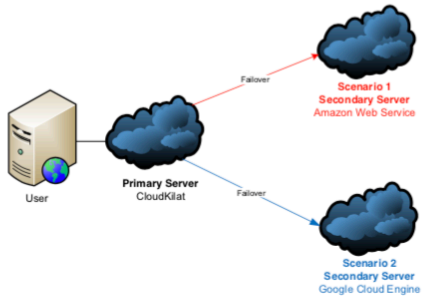


Fig. 4. Implementation Scenario

The second scenario involved another provider that is Google Cloud Engine as secondary system. The workflow of

the system is not much different from another one except different provider that act as secondary server. Furthermore, failover tools used might vary in each scenario since every cloud provider has its own architecture system.

The location of primary and secondary server are different in region and zone. CloudKilat located in Indonesia, while Amazon Web Service and Google Cloud Engine located in United States.

IV. ANALYISIS

In this chapter we analyze several previous study that related with our research particularly on RTO and RPO metrics in experimental testing. Sindhura[14] conducted experimental study between offline tape-based and cloud backup based DRaaS. Tape-based means the mecanism of system backup is saved on tape storage device transferred through LAN media. Otherwise, cloud-based backup transferred via internet connection.

TABLE I RTO and RPO define values[14]

Type of Backup	RTO	RPO
Tape Based	5 to 24 hours	12 hours
Cloud Based	0-4 hours	1 hours

Table 1 represents a very significant different between two model of backup. RTO value in tape-based backup depending on the size of data needed for restoration. Moreover, RPO value in cloud based backup is less than tape-based since the periode of data backup more often.

Suguna classify seven tiers of disaster recovery in relationship with RTO and RPO as revealed in Table 2. The seventh tier is the best choice for lower RTO and RPO values, yet the cost prepared would be increased.

TABLE II Disaster Recovery Level[14]

Tier	Description	RTO	RPO
1	Point in time tape backup	2-7 days	2-24 hrs
2	Tape backup to remote site	1-3 days	2-24 hrs
3	Disk point in time copy	2-24 hrs	2-24 hrs
4	Remote logging	12-24 hrs	5-30 hrs
5	Concurrent ReEX	1-12 hrs	5-30 min
6	Mirrored data	1-4 hrs	0-5 min
7	Mirrored data with failover	0-60 min	0-5 min

As seen in Table 2, this research will implement mirrored data with failover as model for infrastructure. Therefore, the RTO and RPO values for this research are 0-60 minutes and 0-5 minutes range respectively. Since in this research cloud server location might be quite influential, we will consider location as additional parameter that affect RTO and RPO values.

V. DISCUSSION AND CONCLUSION

This research proposed new approach to consider when comparing DRaaS providers in RTO and RPO perspective. Since in previous research several criteria used to contrass DRaaS provider were price, infrastructured, orchestration, location, etc. Comparing cloud to cloud scenario of disaster recovery between different provider is not exist yet, therefore we propose implementation method also RTO and RPO benchmarking analysis between providers as reference for choosing cloud providers. Since this research will build mirrored data with failover based DRaaS, we have stated RTO value is 0-60 minutes and RPO value is 0-5 minutes.

ACKNOWLEDGEMENT

This research was supported by DIKTI (Ministry of Research, Technology and Higher Education of the Republic of Indonesia) under The Fresh Graduate Scholarship Program.

REFERENCES

[1] G. Peter , “DRaaS For Dummies® Veeam® Software Special Edition”, New Jersey:John Wiley & Sons, 2016.

[2] R. Glen, N. Attila, and E. Chris,”Using Amazon Web Services for Disaster Recovery”, available : www.amazonwebservices.com/AWS_Disaster_Recovery.pdf

[3] S. Suguna and A. Suhasini.“Overview of data backup and disaster recovery in cloud”. International Conference on Information Communication and Embedded Systems (ICICES2014) (978) 1–7, February 2014

[4] P. Suraj, M. Sneha, W. Abdul, and S. Sundaram,”Disaster recovery services in the cloud for SMEs,” International Conference on Cloud ComputingTechnologies, Applications and Management (ICCCTAM) , pp.139-144, 8-10 Dec. 2012.

[5] S. Kruti and S. Kavita R, “Online Data Back-up and Disaster Recovery Techniques in Cloud Computing”, International Journal of Engineering and Innovative Technology (IJEIT) Vol 2 Issue 5, pp249-254, November 2012

[6] A. Dario, C. Manuel, H. Ileana, and M. Sergio, “The Use of Cloud Computing in SMEs”, Elsevier International Journal Mathematical and Computer Modelling, Procedia Computer Science 83 (2016) 1207 – 1212, 2016

[7] M. Hossam Abdel Rahman, “A Proposed Model for IT Disaster Recovery Plan”, I.J. Modern Education and Computer Science, pp57-67, April 2014

[8] G. Peter, ”IT Disaster Recovery For Dummies”, Indiana : Wiley Publishing,Inc. 2008.

[9] A. Omar H., M. Yashwant .K, “Assessing Disaster Recovery Alternatives: On-site, Colocation or Cloud “, 2012 IEEE 23rd International Symposium on, Dallas, TX, 27-30, Page(s): 19 – 20, November 2012

[10] P. Manish, L. Seulki, and P. Jong Sou, “Disaster Recovery for System Architecture using Cloud Computing”, 10th IEEE/IPSJ International Symposium on Seoul,Page(s):304–307, 19-23 July 2010

[11] P. Athanasios, and Z. Tomas, “Criticality estimation of IT business functions with the Business Continuity Testing Points method for implementing effective recovery exercises of crisis scenarios”, International Journal of Computer Science Issues, November 2013.

[12] A. Mohammad Matar, A. Ali A, and A. Imad Fakhri, "Data recovery and business continuity in Cloud computing: A Review of the Research Literature", International Journal of Advancements in Computing technology, December 2016

[13] Google Cloud Solution, "Disaster Recovery Cookbook", Available https://cloud.google.com/solutions/disaster-recovery-cookbook.

[14] Y. Shindura, "Effectiveness of Backup and Disaster Recovery in Cloud : Comparative study on Disk and Cloud based Backup and Disaster Recovery". Karlskrona: Blekinge Institute of Technology.

Notifikasi penerimaan paper pada konferensi ICITEE 2018

Teras Teknik Elektro ITS | Webmail ITS | EDAS Login

https://webmail.its.ac.id/src/webmail.php

Current Folder: **INBOX** — Sign Out

Compose | Addresses | Folders | Options | Search | Calendar | Notes | To Do

Message List | Unread | Delete | Previous | Next | Forward | Forward as Attachment | Reply | Reply All | Delete & Prev | Delete & Next

Move to: INBOX — Move

Subject: [ICITEE 2018] Your paper #1570451443 ('Analysis of RTO and RPO of a Service Stored on Amazon Web Service (AWS) and Google Cloud Engine (GCE)')
From: icitee2018-chairs@edas.info
Date: Sun, May 20, 2018 9:05 am
To: "Yorisan Permana Baginda" <yorisan@elect-eng.its.ac.id> (more)
Priority: Normal
Options: View Full Header | View Printable Version | Download this as a file | View Message details | View as plain text

Dear Mr. Yorisan Permana Baginda:

Congratulations - your paper #1570451443 ('Analysis of RTO and RPO of a Service Stored on Amazon Web Service (AWS) and Google Cloud Engine (GCE)') for 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE) has been **accepted**.

(Note: if you submit for the special Journal Track Option, you will receive 2nd email for status of the accepted paper as a journal or conference)

The reviews are below or can be found at https://edas.info/showPaper.php?m=1570451443.

Please make the necessary changes based on reviewers' comments and suggestions. Committee will check whether the revision has been performed or not. Fail to do so, we have a right to exclude your paper from the proceedings.

Please follow the accepted procedures here http://icitee.ugm.ac.id/

Now we would like your cooperation with the double check of your paper.

(0) Please strictly follow the camera ready guidelines for uploading your final manuscript, as explained in http://icitee.ugm.ac.id/camera-ready-guideline.php

(1) For the copyright: Please ensure you process the copyright. The IEEE e-copyright submission can be done in EDAS electronically at 'Copyright form'.

(2) For the paper final version: Please Strictly use and follow to IEEE template Manuscripts

(3) Please ensure the maximum page of your final paper is 6-pages.

(4) All the papers have to go through the file conversion (become PDF file) offered by IEEE PDF eXpress. You can refer to the link here: http://www.pdf-express.org/. You will need the Conference ID to log in, which is: 43364X. After file conversion (become PDF file) offered by IEEE PDF eXpress successfully. You can upload PDF file paper final version in EDAS at 'Final manuscript'

(5) Please take notice that the Camera Ready Paper should be submitted by June 1, 2018.

(6) Each paper should have a registration with Registration Rate before May 25, 2018 (early registration) or June 1, 2018 (late registration). Any paper without registration will be dropped automatically. Please refer to this link. http://icitee.ugm.ac.id/registration.php The registration link and information will be available on May 2018.

(7) With all your cooperation, the presentation schedule would be announced later on.

(8) IEEE reserves the right to exclude a paper from distribution after the conference (e.g. removal from IEEE Xplore) if the paper is not presented at the conference.

We are looking forward to seeing you in Kuta, Bali-Indonesia, on July 24 - 26, 2018.

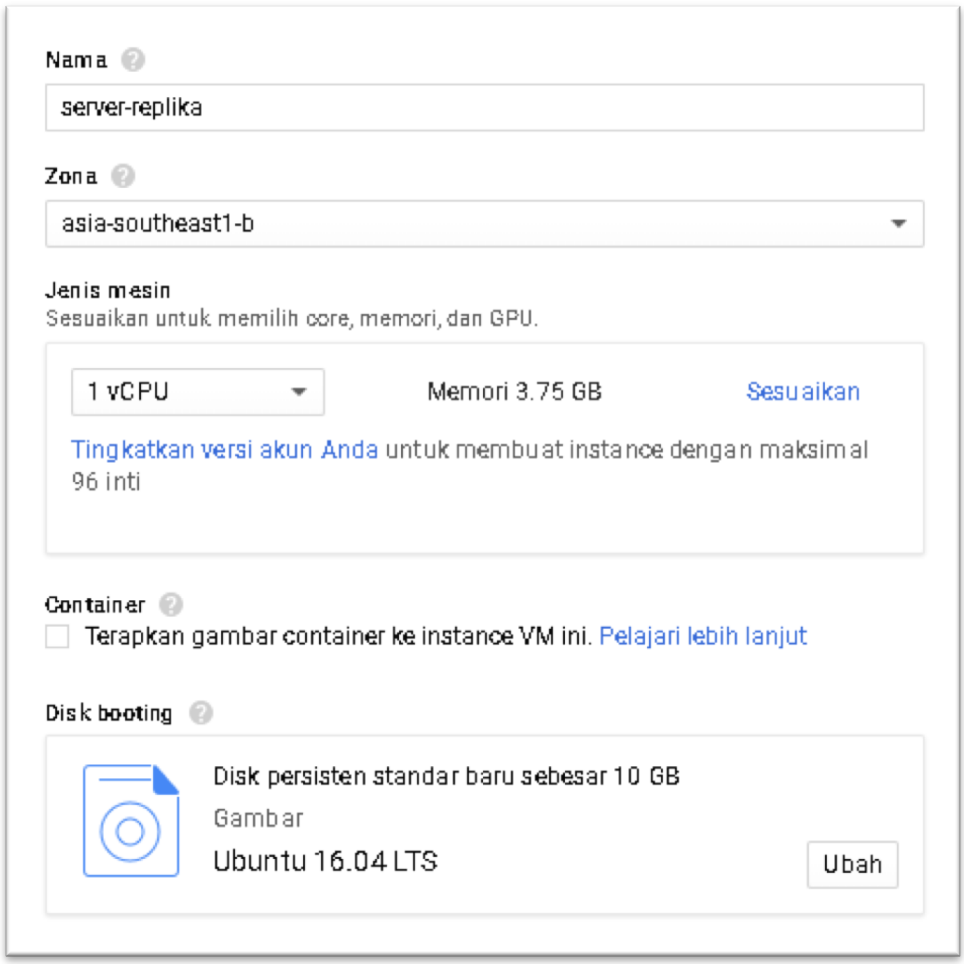
Intalasi VPC pada Google Cloud

Langkah-langkah:

A. Buat intance pada Goole Cloud dengan spesifikasi berikut:

Intance replika	
Jaringan External	Static IP public
Jaringan Internal default	Default
OS	Ubuntu 16.04
Ram	4GB
HA Proxy	
Jaringan External	Static IP public
Jaringan Internal default	Default
OS	Ubuntu 16.04
Ram	4GB

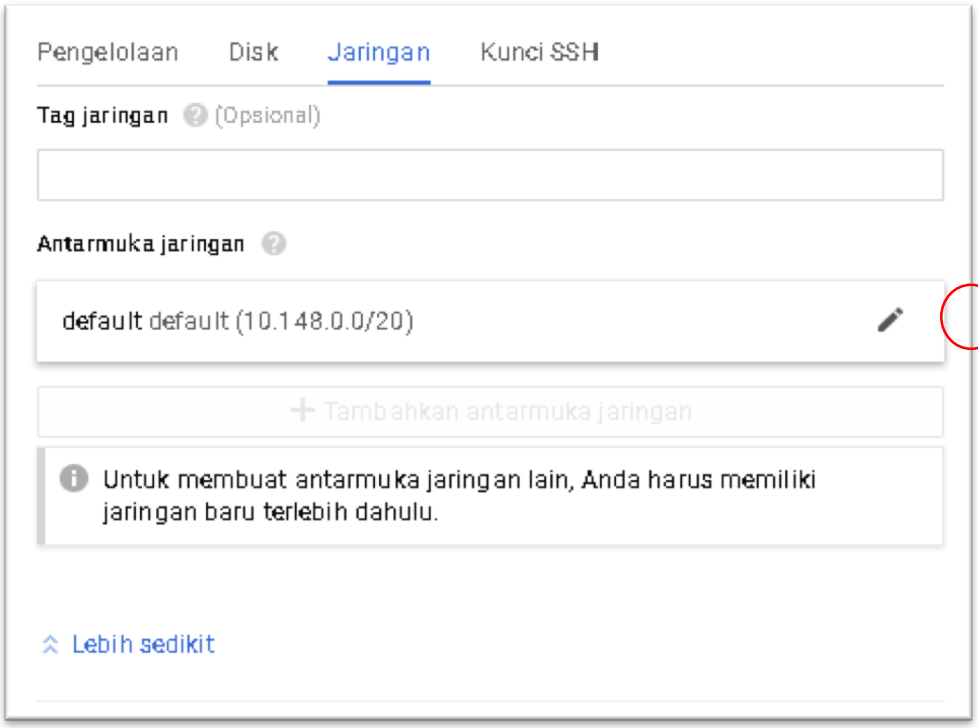
Klik pada pa menu Compute engine untuk membuat instance, kemudian pilih Buat instance

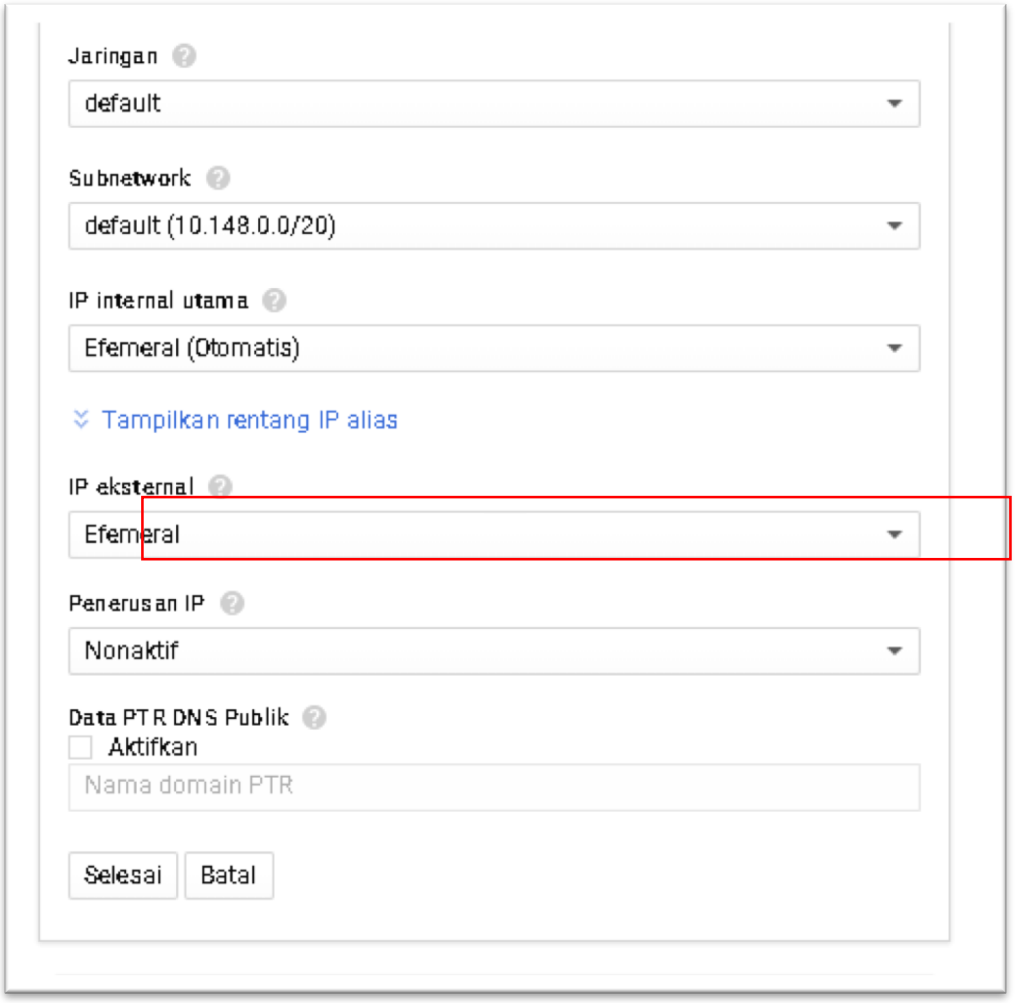


Isi nama instance sesuai dengan kebutuhan.

Checklisht pada izinkan traffic HTTP dan HTTPS

Atur Konfigurasi jaringan untuk mensetting ip static


Pengelolaan
Disk
Jaringan
Kunci SSH
Tag jaringan
(Opsional)
Antarmuka jaringan
default default (10.148.0.0/20)
Tambahkan antarmuka jaringan
Untuk membuat antarmuka jaringan lain, Anda harus memiliki jaringan baru terlebih dahulu.
Lebih sedikit

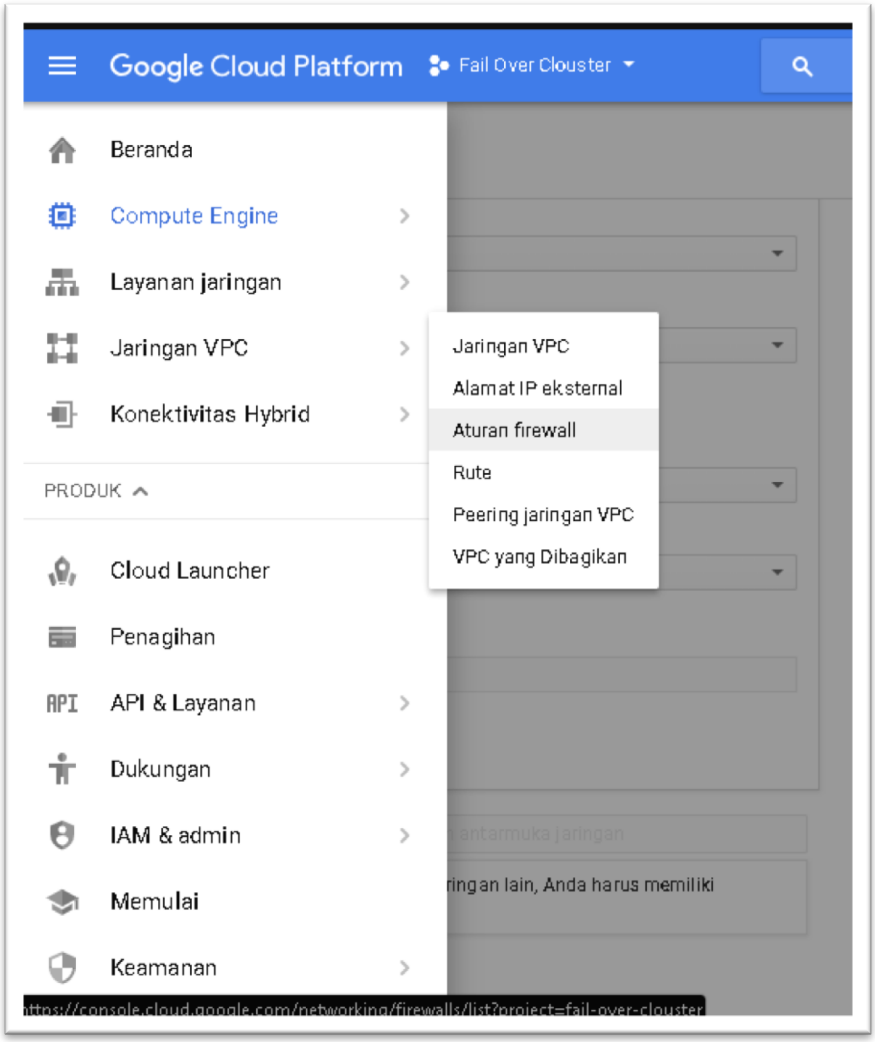

Jaringan
default
Subnetwork
default (10.148.0.0/20)
IP internal utama
Efemeral (Otomatis)
Tampilkan rentang IP alias
IP eksternal
Efemeral
Penerusan IP
Nonaktif
Data PTR DNS Publik
Aktifkan
Nama domain PTR
Selesai
Batal


Pesan alamat IP statis baru
Nama
ipstatik
Deskripsi (Opsional)
BATAL
PESAN

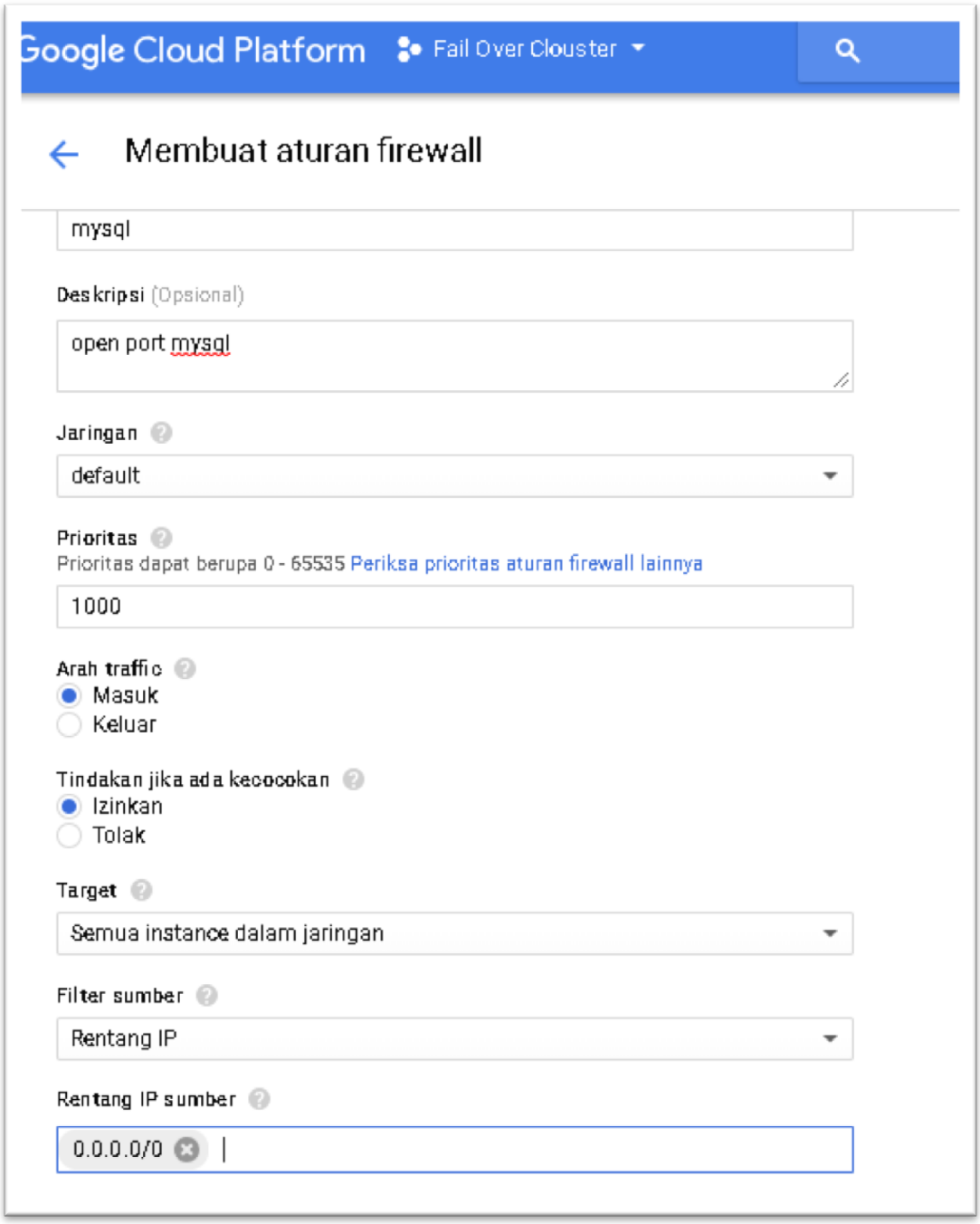
Setting firewall

Setting firewall digunakan untuk membuka port pada server database agar port master pada cloudkilat dapat terintegrasi dengan google cloud.

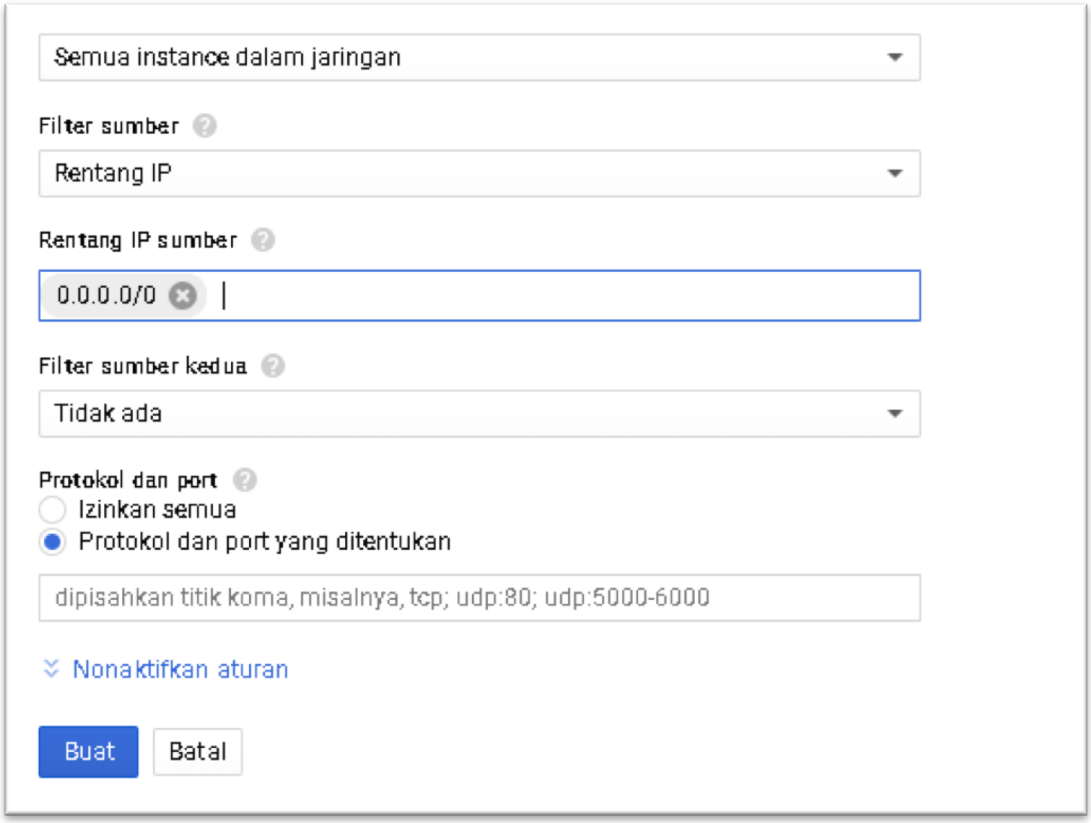
Pilih Jaringan VPC -> Auturan firewall



Isi nama firewall sesuai dengan gambar berikut untuk nama bebas, tapi untuk memudahkan jaringan yang kita buat namakan saja firewallnya mysql



Pada Gambar diatas pilih arah traffic masuk menuju ke server VPC, kemudian isi rentang ip 0.0.0.0/0



Pada protokil masukan port tcp:3306

Intalasi replikas database

CLOUD KILAT

1. apt-get update
2. apt-get install mariadb-server mariadb-client
3. Ubah konfigurasi MariaDB
4.

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

# baris 29, ubah IP bind-address
bind-address = 103.43.45.223

# baris 74, lepas comment
server-id = 1

# baris 75, lepas comment
log_bin = /var/log/mysql/mysql-bin.log
```

5. sudo systemctl restart mysql
6. Buat user untuk akses replikasi

```
mysql -u root -p

GRANT REPLICATION SLAVE ON *.* TO 'replica'@' 35.201.210.241'
IDENTIFIED BY 'secret';
FLUSH PRIVILEGES;
```

7.

8. SHOW MASTER STATUS;

Google Cloud Instance

1. apt-get update
2. apt-get install mariadb-server mariadb-client

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

# baris 29, ubah IP bind-address
#bind-address = 35.201.210.241

# baris 74, lepas comment. Ganti dengan nomor lain, jangan sama dengan
master
server-id = 2

# baris 75, lepas comment
log_bin = /var/log/mysql/mysql-bin.log
```

3. Restart service MariaDB:
   sudo systemctl restart mysql

4. Konfigurasi koneksi ke Master

```
CHANGE MASTER TO

MASTER_HOST='103.43.45.223',

MASTER_USER='replica',

MASTER_PASSWORD='sevret',

MASTER_LOG_FILE='mysql-bin.000002',

MASTER_LOG_POS=9951;
```

5. Jalankan slave

START SLAVE;

6. Jika tidak ada masalah pada saat menjalankan slave, lepas kunci database pada server master
   mysql -u root -p
   UNLOCK TABLES;
7. Tampilkan status slave
8. SHOW SLAVE STATUS \G

Intalasi HAPRoxy

1. apt-get update
2. apt-get install haproxy
3. back up konfigurasi haproxy :

```
cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak
```

4. konfigurasi ha proxy

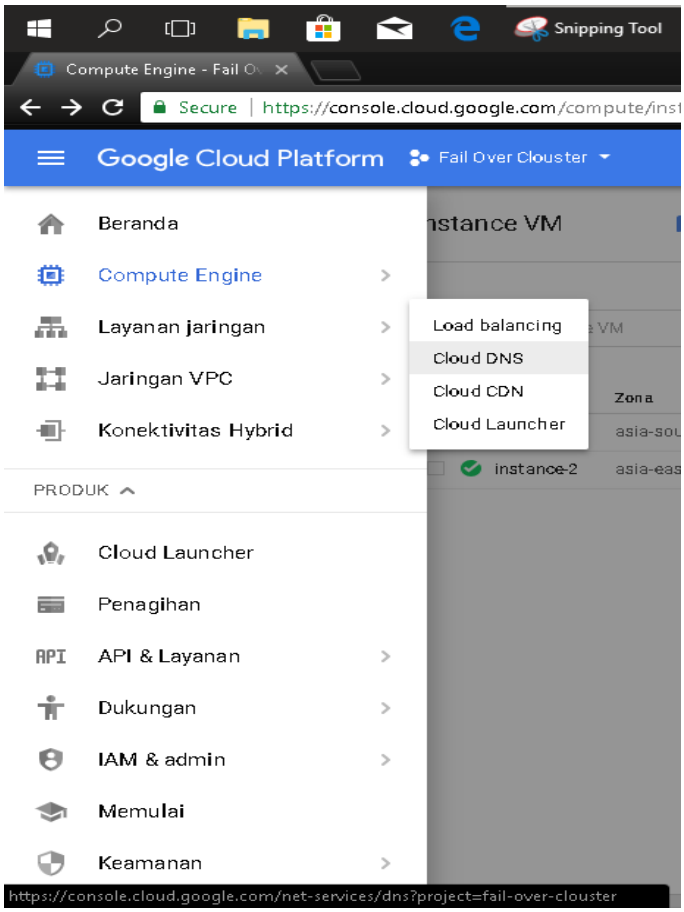
```
frontend LOAD_BALANCER
        bind *:80
        default_backend WEB_SERVER_TIER

backend WEB_SERVER_TIER
        balance roundrobin
        server Webserver1  103.43.45.223:80 check
        server Webserver2  35.201.210.241:80 check

listen stats
    bind :9000
    mode http
    stats enable
    stats hide-version
    stats realm HAproxy-Statistics
    stats uri /haproxy_stats
    stats auth admin:password
```

5. Restar proxy:

```
sudo systemctl restart haproxy
```

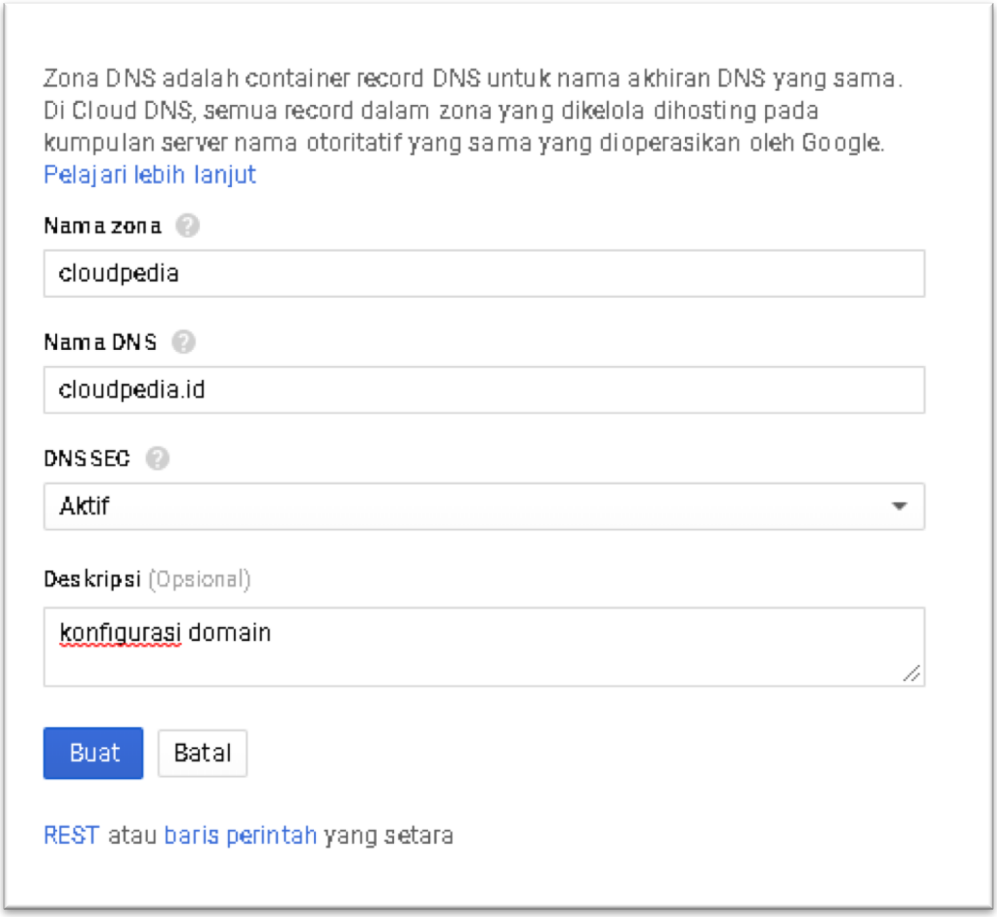
Instalasi Google domain

Pilih Layanan jaringan -> Cloud DNS

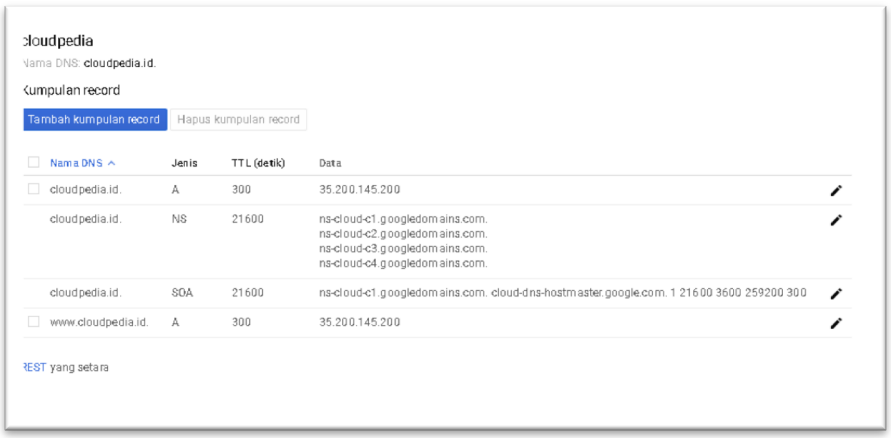


Klik Buat zona

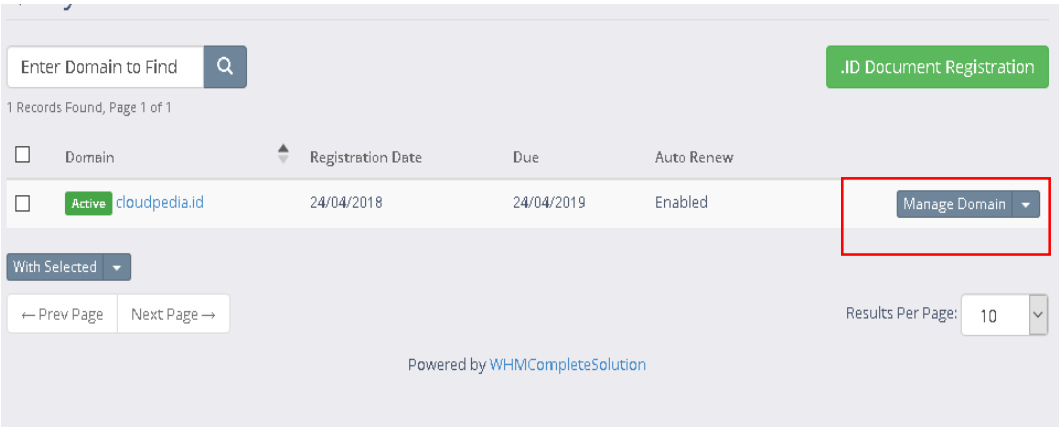
Masukan nama zona sesuai dengan kebutuhan, kemudian isikan nama domain yang telah terdaftar atau yang telah di pesan. Jika sudah klik Buat



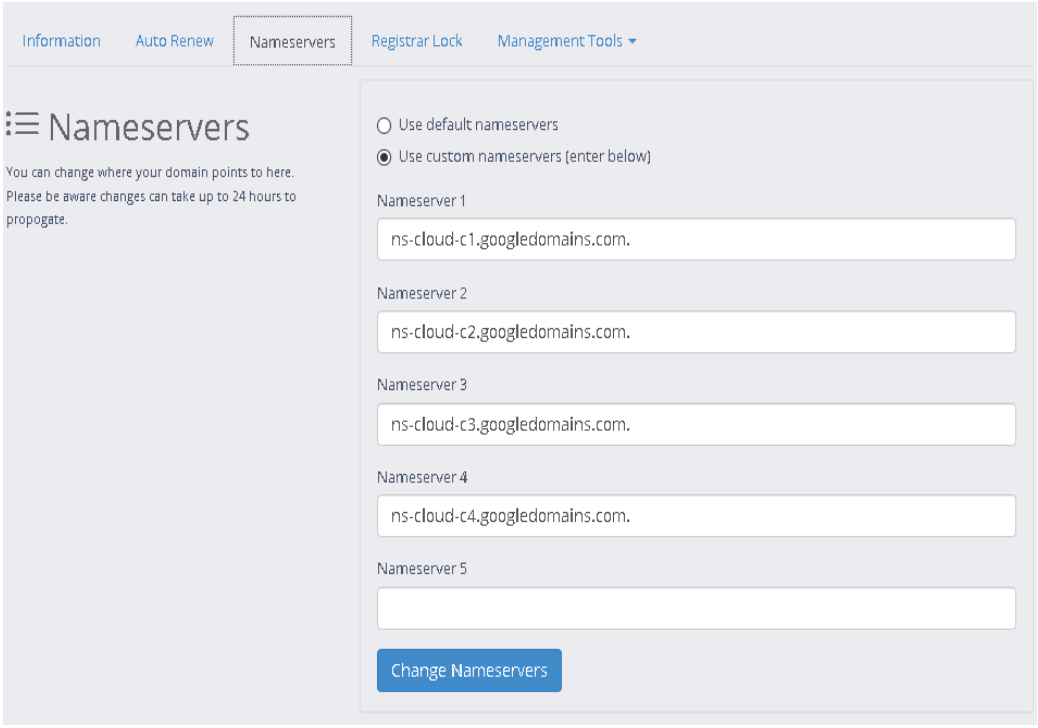
Tambahkan kumpulan record, tujuanya ada merecord ip address pada ip public, ke domain yang ada.



Pada hosting yang cloud yang ada di cloudkilat pilih Manage domain.



Kemudian isikan nama server yang pada pada dns google yang telah terdaftar.



Pengujian

untuk pengujian file over pada server, matikan salah satu node yang ada di cloudkilat atau yang ada di google.

Untuk pengujian repilkasi database.

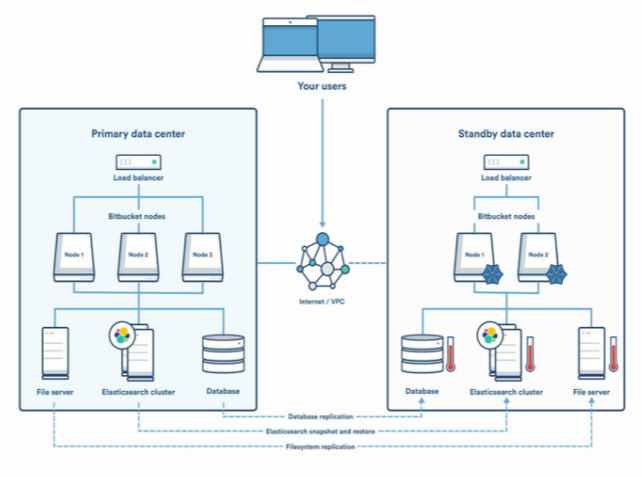
Mysql -u root -p

Masukkan password : hanidalia

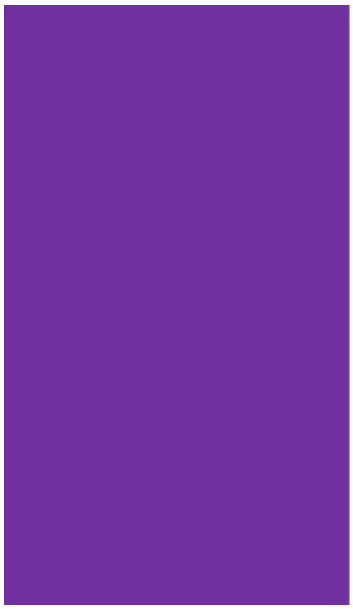
Kemudian masukan command: show databases;

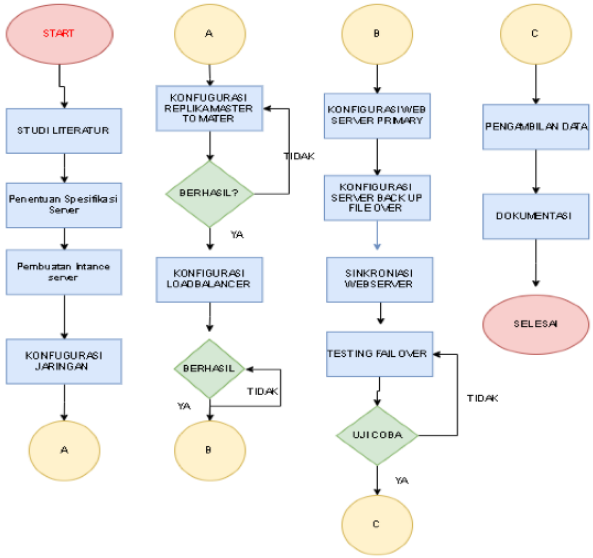
Intalasi VPC pada Amazon Web Service

Langkah 1



langkah 2





langkah 3

PEMBUATAN MASTER NODE CLOUD KILAT

SERVER	IP PUBLIC	PASSWORD/ KETERANGAN	USERNAME
UBUNTU 14.04	103.23.22.76	Wisudaseptember!	Root
Mysql server	0.0.0.0	hanidalia	Root
Apache	103.23.22.76	/var/www/cloudpedia.id/public_html	-

Konfigurasi database server master

Command	Keterangan
sudo apt-get update sudo apt-get upgrade -y sudo apt-get install apache2 php5 php5-mysql mysql-server mysql-client	Intalasi server dan kebutuhan paket
Nano **/etc/mysql/my.cnf**	server_id = 1 log_bin = /var/log/mysql/mysql-bin.log log_bin_index = /var/log/mysql/mysql-bin.log.index relay_log = /var/log/mysql/mysql-relay-bin relay_log_index = /var/log/mysql/mysql-relay-bin.index expire_logs_days = 10 max_binlog_size = 100M log_slave_updates = 1 auto-increment-increment = 2 auto-increment-offset = 1
Setting bind public ip address (untuk public ip)	bind-address = 0.0.0.0
sudo service mysql restart	Restart konfigurasi server

Konfigurasi Database master

Command	Keterangan
mysql -u root -p	Password: hanidalia
GRANT REPLICATION SLAVE ON *.* TO 'replication'@'x.x.x.x' IDENTIFIED BY 'password';	Izin permission server untuk remote
mysql -ureplication -p -h x.x.x.x -P 3306	Testing apakah bisa di remote apakah tidak
SHOW MASTER STATUS;	Pengecekan status log pada server
SLAVE STOP; CHANGE MASTER TO master_host='x.x.x.x', master_port=3306, master_user='replication', master_password='password', master_log_file='mysql-bin.000001', master_log_pos=277; SLAVE START;	Menghubungkan server ke slave
SHOW SLAVE STATUS \G	Pengecekan apakah replika berhasil atau tidak

Konfigurasi apache2

Command	Keterangan
sudo a2dissite *default	Disable aplikasi apache default
cd /var/www	Masuk ke direktori
sudo mkdir cloudpedia.id	Membuat direkrori untuk folder root website
sudo mkdir example.com/public_html sudo mkdir example.com/log	Membuat direktory untuk website
Nano /etc/apache2/sites-available/cloudpedia.id.conf	Lampiran konfigurasi webserver
sudo a2ensite cloudpedia.id.conf	Mengaktifkan konfigurasi baru webserver
sudo service apache2 restart	Restart webserver

Lampiran konfigurasi webserver

```
# domain: example.com
# public: /var/www/example.com/public_html/
<VirtualHost *:80> # Admin email, Server Name (domain name), and any aliases
 Server Admin webmaster@example.com
 Server Name www.cloudpedia.id
 Server Alias cloudpedia.id # Index file and Document Root (where the public files are located)
 index.html index.php
 /var/www/cloudpedia.id/public_html
# Log file locations
 Log_Level warn
  ErrorLog  /var/www/ cloudpedia.id log/error.log
 CustomLog /var/www/ cloudpedia.id log/access.log combined
</VirtualHost>
```

Intalasi Wordpress

Command	Keterangan
cd /var/www	Masuk ke direktory tersebut
wget https://wordpress.org/latest.tar.gz	Download Aplikasi wordpress
tar -xvf latest.tar.gz	Extract file hasil download
cp -R wordpress/* /var/www/cloudpedia.id/public_html	Copy hasil konpresan ke direktory web
mysql -u root -p	Login ke database untuk membuat databse baru
CREATE DATABASE wordpress;	Buat database wordpress
GRANT ALL PRIVILEGES ON wordpress.* TO 'budi'@'localhost' IDENTIFIED BY 'hanidalia';	Buat username baru
FLUSH PRIVILEGES;	Izinkan previlage
EXIT	Keluar dari database
chmod 777 /var/www/cloudpedia.id/public_html/	Berikan permission untuk server dapat di akses

Konfigurasi server replika

Command	Keterangan
sudo apt-get update sudo apt-get upgrade -y sudo apt-get install apache2 php5 php5-mysql mysql-server mysql-client	Intalasi server dan kebutuhan paket
Nano **/etc/mysql/my.cnf**	server_id = 2 log_bin = /var/log/mysql/mysql-bin.log log_bin_index = /var/log/mysql/mysql-bin.log.index relay_log = /var/log/mysql/mysql-relay-bin relay_log_index = /var/log/mysql/mysql-relay-bin.index expire_logs_days = 10 max_binlog_size = 100M log_slave_updates = 1 auto-increment-increment = 2 auto-increment-offset = 2
Setting bind public ip address (untuk public ip)	bind-address = 0.0.0.0
sudo service mysql restart	Restart konfigurasi server

Konfigurasi database replika

Command	Keterangan
mysql -u root -p	Password: hanidalia
GRANT REPLICATION SLAVE ON *.* TO 'replication'@'x.x.x.x' IDENTIFIED BY 'password';	Izin permission server untuk remote
mysql -ureplication -p -h x.x.x.x -P 3306	Testing apakah bisa di remote apakah tidak
SHOW MASTER STATUS;	Pengecekan status log pada server
SLAVE STOP; CHANGE MASTER TO master_host='x.x.x.x', master_port=3306, master_user='replication', master_password='password', master_log_file='mysql-bin.000001', master_log_pos=277; SLAVE START;	Menghubungkan server ke slave
SHOW SLAVE STATUS \G	Pengecekan apakah replika berhasil atau tidak

Sinkroniasi webserver master replika

Command	Keterangan
rsync -avL --progress -e "ssh -i ghost.pem" \ /var/www/* \ root@eec2-18-188-58-106.us-east-2.compute.amazonaws.com:/var/www/.	Remote agara file tersebut dapat singkron ke server utama. Pembuatan server folder mengikuti cara server master.
sudo service apache2 restart	Restart server

Instalasi load balancer

Command	Keterangan
Apt-get install haproxy	Install load balancer
Nano /eta/haproxy/<file konfigurasi>	LAmpiran

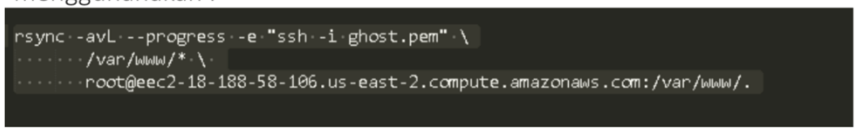
Lampiran konfigurasi

```
frontend LOAD_BALANCER
        bind *:80
        default_backend WEB_SERVER_TIER

backend WEB_SERVER_TIER
        balance roundrobin
        server Webserver1  103.23.22.76:80 check
        server Webserver2  18.220.46.86:80 check

listen stats
    bind :9000
    mode http
    stats enable
    stats hide-version
    stats realm HAproxy-Statistics
    stats uri /haproxy_stats
    stats auth admin:password
```

Uji coba

1. Akses webserver pada url: http://ec2-18-219-189-47.us-east-2.compute.amazonaws.com/
2. Login dengan Username: seperadmin Password: Wisudaseptember!
3. matikan salah satu node, dimaster atau di slave (Amazon)
4. posting beberapa postingan.

Keterangan

1. Untuk perubahan tampilan sinkronisasi server harus dilakukan dengan menggunanakan :
2. 

Total Cost of Ownership

No	Kebutuhan	Biaya/bulan(Rp)
Skenario 1(Cloudkilat ke Google Cloud)		
1	1 unit VPC(Virtual Private Server) Cloudkilat	203.500
2	1 unit domain	200.000
3	2 unit VPC (Virtual Private Server) Google Cloud	-
Total		403.500
Skenario 2 (Cloudkilat ke Amazon Web Service)		
1	1 unit VPC(Virtual Private Server) Cloudkilat	203.500
2	2 unit VPC (Virtual Private Server) Amazon Web Service	-
Total		203.500