Sinergi Replikasi Server dan Sistem *Failover* pada *Database Server* untuk Mereduksi *Downtime Disaster Recovery Planing* (DRP)

Wahyu Ari Yuliono[1], Agus Prihanto[2]
[1,2] Teknik Informatika, Universitas Negeri Surabaya
[1]wahyu.18079@mhs.unesa.ac.id
[2]agusprihanto@unesa.ac.id

Abstrak*—Data tidak hanya digunakan sebagai media untuk pertukaran informasi melainkan juga sebagai alat komunikasi antar perangkat yang sudah diintegrasikan. Data menjadi kebutuhan yang penting, tidak terbayangkan bagaimana data hilang atau tidak dapat diakses lagi. Beberapa sebab data tidak dapat diakses seperti data *corrupt* atau bisa juga data mengalami gangguan saat akan diakses, akses ganguan bisa berasal dari koneksi jaringan bisa juga berasal dari server dari data tersebut. Bahaya yang paling besar adalah bencana. Jika bencana tersebut terjadi menyebabkan banyak data hilang, maka dari itulah *Disasster Recovery Planning* (DRP) diperlukan. Waktu *downtime* menjadi penentu apakah suatu DRP efektif atau tidak. Sehingga perlu untuk mereduksi waktu dari *downtime* tersebut. Selain waktu downtime permasalahan ketersediaan data pada server erat kaitannya dengan *high availability* dengan protokol heartbeat. Protokol heartbeat digunakan untuk sistem failover dari server webserver. Selain *failover* webserver, proses *backup* dan *restore* database server melibatkan backup dari database server utama, untuk restore akan menggunakan *differential* backup. Sinkronisasi server dilakukan dengan rsync. Perintah rsync dikombinasikan dengan crontab untuk proses penjadwalan sedangkan proses restore pada server backup bergantung pada waktu pemindahan data dari server utama dan proses backup pada server utama itu sendiri. Hasil pengujian menujukkan bahwa sinergi dari sistem failover dan replikasi server mereduksi waktu downtime untuk link server efektif diterapkan pada suatu DRP dengan mereduksi waktu downtime jika dibandingan dengan DRP yang dilakukan secara manual.

Kata Kunci—**DRP, Database server, Heartbeat, Replikasi server.**

I. PENDAHULUAN

Data merupakan kumpulan informasi yang memuat banyak hal. Penyimpanan data juga berbagai macam mulai dari *hardisk*, *flashdisk*, maupun secara *cloud*. Pada era modern seperti ini, data sangat penting dalam suatu bisnis maupun dibidang lainnya. Data tidak hanya digunakan sebagai media untuk pertukaran informasi melainkan juga sebagai alat komunikasi antar perangkat yang sudah diintegrasikan. Data menjadi kebutuhan yang penting, tidak terbayangkan bagaimana data hilang atau tidak dapat diakses lagi. Hal ini akan menjadi suatu masalah yang besar bahkan penting bila data tersebut merupakan data yang memuat informasi *realtime* yang digunakan saat itu juga.

Beberapa sebab data tidak dapat diakses seperti data *corrupt* atau bisa juga data mengalami gangguan saat akan diakses, akses ganguan bisa berasal dari koneksi jaringan bisa juga berasal dari server dari data tersebut. Bahaya yang paling besar adalah bencana. Bencana tidak dapat diprediksi dengan tepat, seperti: kebakaran, kebanjiran, ataupun bencana alam lain akan membahayakan data yang tersimpan pada suatu database server. Jika bencana tersebut terjadi menyebabkan banyak data hilang, maka dari itulah *Disasster Recovery Planning* (DRP) diperlukan untuk memulihkan / *recovery* data sebagai langkah antisipasi kehilangan data atau kerusakan data. Beberapa jurnal penelitian telah memuat perencanaan ini [1] menggunakan framework NIST SP 800-34 dengan mengidentifikasi kebutuhan yang krusial pada perusahaan dan menerapkan langkah – langkah sesuai dengan kerangka kerja tersebut. DRP juga menjadi solusi untuk melindungi arsip [2] yang di terapkan pada BPN di daerah D.I. Yogyakarta. Inti dari DRP adalah pada *backup* data dari suatu server serta waktu yang diperlukannya untuk bisa kembali lagi *standby*, metode *backup* database ada berbagai macam, salah satunya *Hot Standby* [3] metode dengan menduplikasi server induk dengan server cadangan, sehingga struktur pada database tersebut sama.

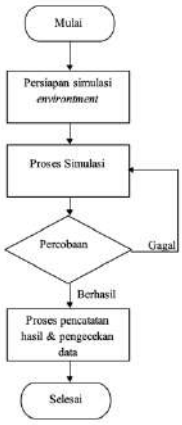
DRP tidak hanya fokus pada back up data saja namun hal yang terpenting lainnya adalah waktu. Skala waktu untuk pemulihan terdiri dari *Respone Time Objective* (RTO) dan *Respon Point Objective* (RPO). RTO adalah waktu toleransi untuk gangguan yang terjadi sedangkan RPO adalah toleransi banyaknya data yang hilang [4]. *Downtime* yaitu kondisi dimana kejadian gangguan itu terjadi. Waktu *downtime* bervariasi tergantung pada data yang akan dipulihkan, semakin besar data yang dipulihkan maka akan membutuhkan waktu yang lumayan lama. Waktu *downtime* inilah yang menentukan suatu DRP efektif atau tidak. Perlu untuk mereduksi waktu yang diperlukan untuk mengembalikan server kembali *stand by* sehingga waktu *downtime* semakin sedikit dan suatu DRP bisa menjadi efektif saat bencana benar – benar terjadi. Selain waktu yang dikurangi data juga perlu untuk dipulihkan dengan cepat dan tepat.

Pemulihan data tidak selamanya data yang dipulihkan selalu sama dengan data yang lama, bisa terjadi beberapa masalah seperti terdapat data yang hilang dan waktu untuk pemulihan data yang lama. Apalagi aspek *cost* untuk pemulihan data juga menjadi aspek untuk dipertimbangkan dalam suatu bisnis dan juga DRP dari suatu perusahaan. Pemulihan data saat DRP bisa dilaksanakan dengan lebih efisien dengan adanya deteksi dini, hal ini akan bisa mengetahui status dari server itu sendiri. *Failover* digunakan untuk mendeteksi dini server dan pernah diterapkan pada suatu server, seperti pada penelitian clustering pada server Windows 2012-R2 menggunakan hyper V [5]. Pada sistem operasi linux sistem *failover* juga dapat dilakukan sama halnya seperti pada windows server. Sinergi dari failover dan *backup* database menjadi salah satu pilihan dalam penyusunan suatu DRP baik perusahaan ataupun pada instansi dalam menjaga datanya.

Pada penelitian ini menerapkan DRP dengan sinergi sistem replikasi dan sistem failover. Sistem replikasi adalah mereplikasi server backup sehingga sama dengan server utama untuk segala aspek, sedangkan sistem failover sendiri adalah pendeteksi dini bila terjadi ganguan pada node server tersebut. Sinergi dari replikasi server dan sistem failover nantinya akan mereduksi downtime dari DRP yang diterapkan, dengan memindahkan node secara otomatis dan data yang dibackup secara berkala dengan prinsip replikasi diharapkan waktu *downtime* berkurang dan pemulihan data dapat berlangsung dengan relatif cepat. Sistem replikasinya menerapkan backup *deferential* dan sistem failover menggunakan *heartbeat* untuk kedua sisi server yang telah dikonfigurasikan terlebih dahulu.

II. METODOLOGI PENELITIAN

Pada penelitian ini semua dilakukan melalui simulasi dengan tidak menggunakan server asli melainkan menggunakan virtualisasi di GNS3. Simulasi diawali dengan mempersiapkan beberapa software yang diperlukan diataranya GNS3 yang digunakan sebagai media untuk membuat jaringan , Virtualbox yang akan digunakan sebagai media server simulasi. Berikut ini alur dari rencana penelitian.



Gbr. 1 Alur Penelitian

A. *Pembuatan Lingkungan Simulasi*

Proses simulasi suatu jaringan memerlukan beberapa komponen mulai dari server yang disimulasikan maupun perangkat jaringan yang akan digunakan seperti OS (Operting system), router, ataupun hub. *Software* untuk simulasi saat ini sangat banyak yang paling sering digunakan adalah GNS3 serta Cisco packet tracer, khusus untuk cisco packet tracer hanya mensimulasikan perangkat khusus cisco. GNS3 lebih banyak memberikan pilihan, selain menggunakan cisco juga bisa menggunakan perangkat lain contohnya mikrotik. Untuk penelitian kali ini menggunakan GNS3 dengan versi terbaru yakni versi 2.2.20. Selain itu diperlukan beberapa *virtual server*, untuk menambahkannya diperlukan *software* tambahan virtual box. Operating System yang digunakan untuk simulasi adalah Ubuntu 18.04 LTS kemudian dilakukan upgrade menjadi versi 20.04 LTS. OS pendukung untuk kontrol database menggunakan Windows Edge Dev Version, dengan masa evaluation mencapai 120 hari. Selanjutnya OS dibagi menjadi beberapa server dengan tugas masing-masing , untuk detail pada table dibawah ini.

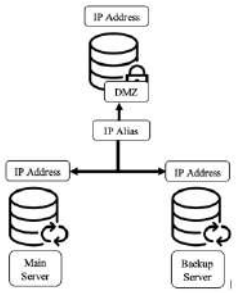
TABEL I
DATA SERVER SIMULASI

Server	OS	Fungsi	Hostname
Server Utama	Ubuntu 20.04 LTS	Main server	Ubuntusu
Server Backup	Ubuntu 20.04 LTS	Secondary server	Serverbu
DMZ	Ubuntu 20.04 LTS	DMZ server	Ubuntudmz
Server kontrol	Windows 10 Edge DEV	Maintenance database	windows

Selanjutnya pada Server tersebut juga dilakukan instalasi dari aplikasi yang dibutuhkan, untuk server utama dan server *backup* dilakukan instalasi aplikasi Microsoft SQL Server, SSH, Heartbeat, dan Rsync. Aplikasi untuk DMZ adalah apache sebagai web server, PHP 7.4, SSH, ODBC, SQLCMD dan SQLSRV. Sedangkan untuk Server kontrol ada aplikasi SSMS (SQL Server Management Studio) versi 18.

B. *High Availability*

Penelitian kali ini membahas tentang *downtime* artinya erat kaitannya dengan ketersedian *website* yang selalu dapat diakses kapanpun dan dimanapun, sangat beresiko jika *website* tersebut mengalami *error* atau tidak dapat diakses ini lah fungsi dari *high availability* yang perlu ada pada system dari website tersebut. Penyimpanan data pada system tersebut didistribusikan pada bebeberapa node, sehingga dapat menghindari apabila ada node yang offline dan dapat berpindah ke node yang online, server yang berkorelasi adalah *web server* dan *database server* [6]. Protokol heartbeat bisa diimplementasikan sebagai upaya untuk memenuhi *high availability* suatu *website*. Berikut ini gambaran protokol heartbeat pada suatu server



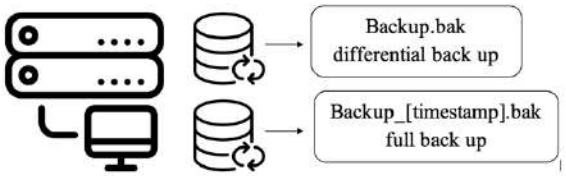
Gbr. 2 Protokol Heartbeat

Protokol *Heartbeat* bekerja untuk jaringan yang akan mengkomunikasikan status server antara node A yang berfungsi sebagai *main* server dan node B sebagai *backup* server saling terhubung dengan management node (MGM Node). Ketika node yang mengalami gangguan kemudian tersebut sudah diperbaiki dan kembali hidup maka proses migrasi kembali dilakukan untuk mengembalikan ke main server atau node asalnya [7]. Kedua server dihubungkan

menggunakan IP alias dimana IP alias ini lah yang diakses oleh *webserver* atau DMZ (Demilitazed Zone), sehingga saat terjadi gangguan, user bahkan tidak mengetahui terjadi gangguan karena proses pemindahan node secara otomatis yang sudah dipetakan dengan menggunakan protocol heartbeat. Untuk protokol heartbeat sendiri saat ini sudah terdapat 2 versi, versi 1 hanya mendukung 2 node, sedangkan versi terbarunya mendukung lebih dari 2 node dengan konfigurasi yang berbeda juga dari versi 1.

C. *Backup dan Restore Database*

Proses *backup* ataupun *restore* pada suatu database sudah wajar dilakukan, baik untuk keperluan pemindahan database, untuk update aplikasi, untuk migrasi server, ataupun pengambilan data agar tidak mengganggu proses yang ada disuatu perusahaan. Contoh saat audit dilakukan dengan mengecek data pada suatu database dengan periode tertentu sehingga dilakukan backup database dan restore pada server yang sudah ditentukan. Database memiliki beberapa jenis, seperti MySQL, MSSQL, Postgree, Oracle Database, sedangkan layanan cloud juga saat ini telah banyak tersedia contohnya Microsoft Azure dan Amazon Web Service (AWS). Penelitian kali ini menggunakan MSSQL atau biasa dikenal dengan Microsoft SQL Server. Pemilihan database ini sebagai implikasi dari beberapa perusahaan yang masih masiv menggunakan database ini. Selain itu database yang dikembangkan oleh Microsoft ini memiliki beberapa keunggulan, diantara adanya fitur *backup* otomatis menggunakn SQL Agent atau bisa menggunakan Jobs yang sudah dilakukan konfigurasi terlebih dahulu. Selain *backup* otomatis juga tidak lupa ada fitur backup manual yang bisa dilakukan dengan mengakses menu *task* selanjutnya pilih *backup* atau *restore*. Pada *backup* database SQL server ini bisa dilakukan back up full artinya backup dilakukan dengan membackup seluruh data yang ada mulai dari database di inisialisasi sedangkan terdapat *differential*/*incremental backup* dimana *backup* ini hanya meliputi tambahan data dari database tersebut. Kedua metode backup database ini sudah meliputi transaksi dan juga log pada database yang akan dibackup



Gbr. 3 Detail backup database SQL server

Proses restore database SQL server melibatkan backup dari database yang sudah dilakukan, untuk restore akan menggunakan database dengan defferntial backup diharapkan proses restore akan belangsung dengan cepat, dengan asumsi database sudah di initialisasi sama antara main server dengan backup server. Database dengan full backup sebagai backup dari database nantinya dilakukan migrasi setelah DRP atau pun jika gagal sinkronisasi dan restore antara server backup dengan main server

D. *Sinkronisasi Data Server*

Sinkronisasi server diperlukan untuk pemindahan data hasil backup dari server main ke server backup, aplikasi yang cocok untuk OS Ubuntu / Linux adalah Rsync (Remote Sync). Rsync memungkinkan transfer file incremental dengan hanya mentransfer perbedaan antara sumber dengan lokasi tujuan. Rsync biasa digunakan dalam hal mirroring data, backup ataupun menyalin file antara system. Hal inilah menjadikan rsync cocok untuk digunakan dalam replikasi server pada DRP kali ini. Selain itu Rsync juga menggantikan perintah pada linux seperti scp, sftp dan cp. Berikut ini perintah dasar rsync :

L ke L : rsync [OPT]…SRC[SRC]...DT
L ke R : rsync [OPT]…SRC[SRC]…[U]@H:DT
R ke L : rsync [OPT]…SRC[SRC]…[U]@H:SRC[DT]

Keterangan :

L : local
R : remote
OPT : rsync options
SRC : source
DT : destination (tujuan)
U : user
H : remote host

Selain itu rsync juga menyediakan beberapa option antara lain :

-a , --archive, -r, -rlptgoD untuk sinkronisasi folder
-z, --compress untuk mengkompres data saat dikirim
-p, --partial –progress untuk menampilkan progress transfer
-- delete untuk menghapus file asing di lokasi tujuan
-q, --quiet untuk output pesan menampilkan error
-e untuk memilih remote shell yang berbeda

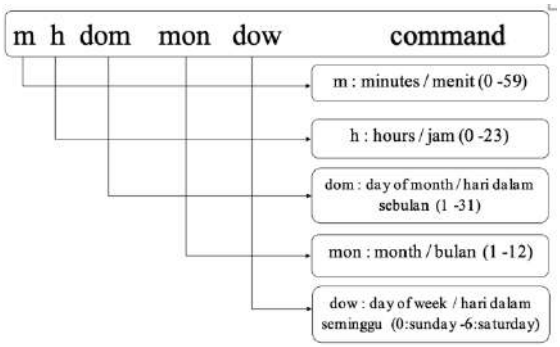
Contoh sederhana perintah rsync :

```
User# rsync -a /opt/namefile.zip /temp/
```

Artinya memindahkan file namefile.zip ke dalam folder temp.

E. *Penjadwalan dengan Crontab*

Sinkronisasi server telah dilakukan dengan rsync namun perintah rsync ini masih harus dijalankan secara manual, untuk membuatnya menjadi otomatis makan perlu dilakukan *schedule*/penjadwalan. Pada OS Windows penjadwalan dengan perintah digunakan dengan *scheduler* dan perintah disimpan dalam bentuk .bat sehingga nantinya akan diakses dengan *scheduler*. Pada OS Ubuntu fitur penjadwalan ini juga ada yaitu crontab. Crontab sendiri lebih simple dibandingkan dengan scheduler pada Windows, berikut ini penjelasan dasar dari perintah crontab pada Ubuntu.



Gbr. 4 Perintah dasar crontab

Selanjutnya perintah dari rsync akan dikombinasikan dengan crontab seperti ini :

*/30 * * * * rsync -r path/source/*.pdf path/ destination

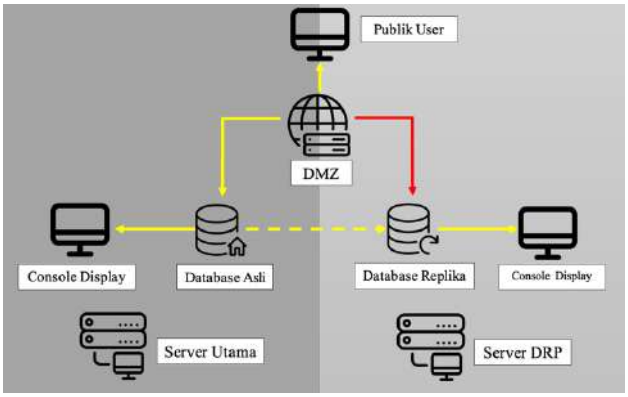
Dari perintah diatas dapat diartikan perintah rsync akan diulang setiap 30 menit, untuk perintah sinkronisasinya sendiri melakukan pemindahkan setiap file dengan ekstensi pdf dari folder source ke folder destination. Rsync hanya akan memindahkan file yang tidak ada sebelumnya jadi walaupun perintah diulang setiap 30 menit jika tidak ada data baru maka tidak akan dilakukan pemindahan data.

F. *Skenario DRP*

Setelah melakukan semua persiapan yang diperlukan, maka skenario DRP bisa dilakukan. Skenario meliputi kondisi normal maupun penanganan *error*, tapi tidak mencakup migrasi kembali setelah melakuan DRP artinya proses hanya berjalan satu arah saja tanpa diikuti dengan proses pengembalian data ke main server. Berikut ini detailnya :

1) Skenario DRP Kondisi Normal

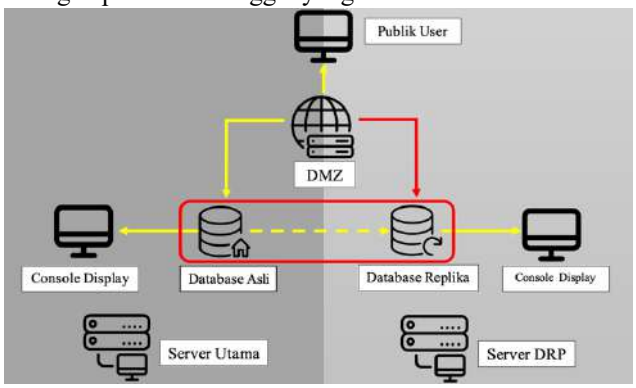
Pada kondisi normal, kedua server berjalan secara bersamaan. Server utama (main server) dan server replikasi (server backup) DRP akan diinisialisasi secara sama pada saat pertama – tama, sehingga dilakukan backup full pada database. Selanjutnya akan dilakukan backup secara berkala. Kedua konfigurasi server sama sehingga tidak ada perbedaan antara kedua server.



Gbr. 5 Kondisi normal

2) Skenario DRP Kondisi Pemulihan

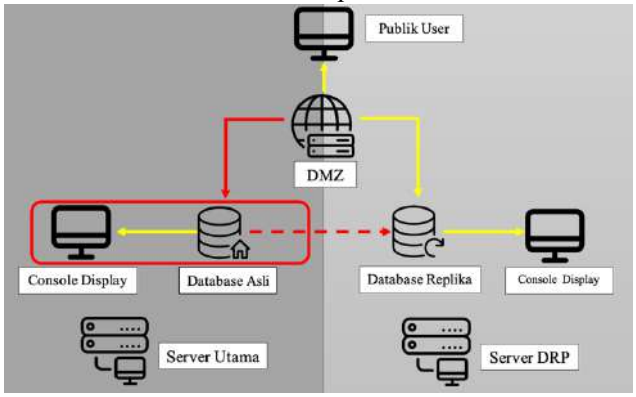
Pemulihan atau *backup* dilakukan secara berkala dengan menggunakan *scheduler* jobs SQL Server. Selanjutnya hasil backup akan dipindahkan secara berkala dengan metode Rsync ke backup server, sehingga setiap harinya akan ada backup database baru dan backup yang lama akan digantikan (reduce storage issue) dengan ketentuan database backup lebih dari 30 hari selain itu akan digantikan dengan backup database yang baru dengan penamaan tanggal yang sama.



Gbr. 6 Kondisi pemulihan

3) Skenario DRP Kondisi Bencana

Pada saat kondisi bencana dimana link pada server utama kemungkinan putus (tidak dapat diakses) sehingga heartbeat yang berfungsi sebagai pendeteksi kondisi server akan otomatis memindahkan link server ke backup server DRP.



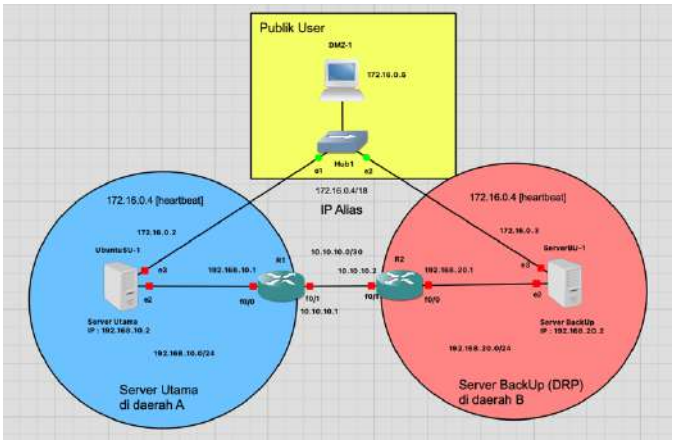
Gbr. 7 Kondisi pemulihan

III. HASIL DAN PEMBAHASAN

Saat seluruh persiapan DRP simulasi telah dilakukan, maka skenario pengujian DRP bisa dijalankan. Skenario pengujian DRP pada simulasi ini tergantung pada perangkat keras yang menjalankan simulasi tersebut. Pada penelitian ini DRP simulasi di jalankan pada komputer atau laptop dengan spesifikasi sebagai berikut :

Processor	: 2,7 GHz Dual-Core Intel Core i5
RAM	: 8 GB 1867 MHz DDR3
Storage	: 256 GB SSD
GNS3	: 2.2.20
Virtualbox	: 6.1.22 r144080

Berdasarkan spesifikasi yang sudah disebutkan, laptop tersebut hanya mampu menjalankan 3 VM (*Virtual Machine*) sehingga saat pengujian dilakukan secara manual dan otomatis bergantian. Pada pengujian secara manual untuk melakukan pengecekan apakah konfigurasinya sudah benar, sedangkan pengujian otomatis untuk mengukur waktu yang diperlukan DRP berjalan. Topologi penelitian ini pada simulasi GNS3 bisa dilihat sesuai dengan gambar di bawah. Diasumsikan baik server utama dan server backup berada dilingkungan yang berbeda pada kenyataannya dengan adanya router yang berbeda jaringan. Simulasi ini diharapkan sesuai dengan kondisi nyata pada lapangan.



Gbr. 8 Topologi simulasi DRP

Pada gambar topologi DRP dibagi menjadi 3 bagian, pertama bagian main server yang merupakan server utama (daerah warna biru) dengan hostname ubuntusu, kedua backup server merupakan server replikasi (daerah warna merah) dengan hostname serverbu, dan bagian ketiga DMZ yang digunakan sebagai webserver (daerah warna kuning) dengan hostname ubuntudmz. DMZ berfungsi juga sebagai komputer publik user untuk mengecek koneksi database nantinya. Rincian dari IP address masing – masing perangkat pada tabel berikut.

TABEL III
DATA IP ADDRESS PERANGKAT SIMULASI

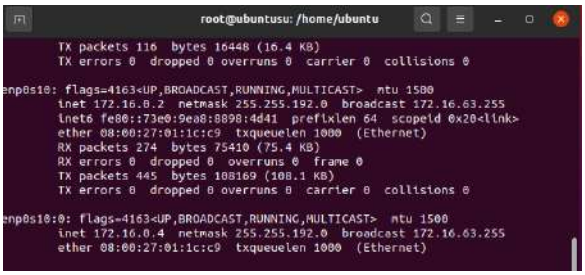
Perangkat	IP address/CIDR	Interfaces	Fungsi
Server Utama (ubuntusu)	172.16.0.2 /18	e3	link ke DMZ
	192.168.10.2 /24	e2	link ke Router1
	172.16.0.4	e3	IP alias
Server Backup (serverbu)	192.168.20.2 /24	e2	link ke Router2
	172.16.0.3 /18	e3	link ke DMZ
	172.168.0.4 /18	e3	IP alias
DMZ	172.16.0.5 /18	e3	ke server Database
Router1	192.168.10.1 /24	f0/0	link ke server utama
	10.10.10.1 /24	f0/1	link ke Router2
Router2	10.10.10.2 /24	f0/1	link ke Router1
	192.168.10.1 /24	f0/0	link ke server backup

A. *Skenario pengujian high availability*

Pada pengujian *high availability* dengan heartbeat pengujian dilakukan dengan cek ip pada masing-masing server terlebih dahulu serta DMZ. DMZ terkoneksi dengan server database melalui IP alias sehingga saat node satu mati otomatis heartbeat akan melakukan konfigurasi dan memindahkan node lain yang masih aktif. Dalam penelitian ini node utama adalah ubuntusu dan node cadangan serverbu. Perpindahan note ini ditandai dengan adanya *interfaces cloning* dibawah *interfaces* yang asli dari server tersebut. Berikut ini hasil dari pengujian yang dilakukan.

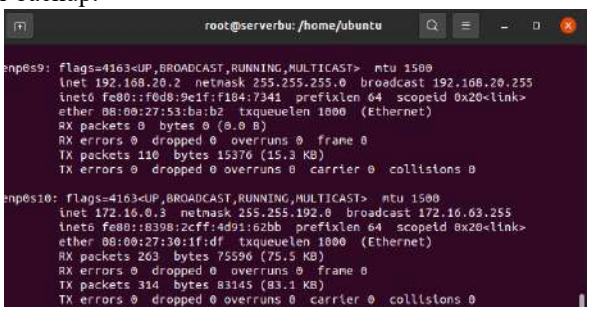
Gbr. 9 Heartbeat aktif pada server

Saat heartbeat aktif pada server maka heartbeat akan selalu melakukan pengecekan pada setiap node yang sudah dikonfigurasikan sebelumnya, beikut ini tampilan dari *interfaces* server simulasi.



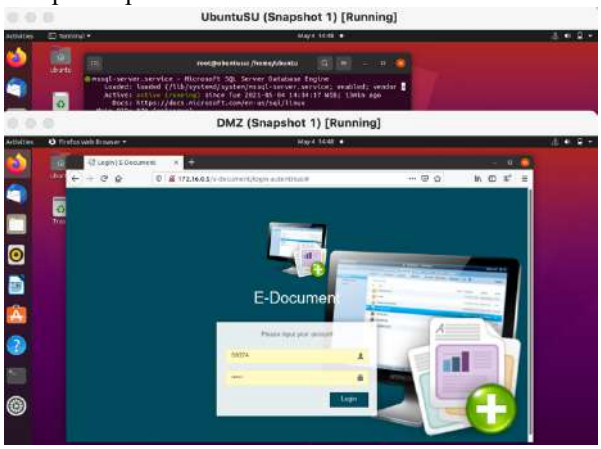
Gbr. 10 IP address pada server utama

Pada gambar diatas telihat ada perbedaan dimana ada tambahan IP address enp0s10:0 ini merupakan IP alias dari heartbeat yang saat ini berada dan aktif pada server utama. Bila node ini mengalami gangguan atau mati maka heartbeat akan memindahkan ke node yang sudah ditentukan ditandai dengan adanya ip tambahan seperti tadi. Berikutnya IP address dari server backup.



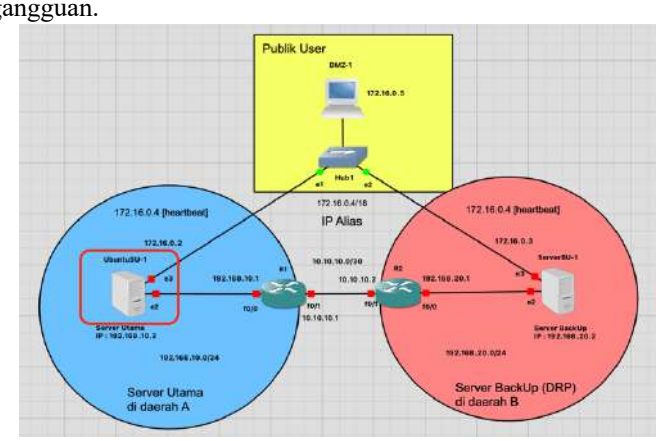
Gbr. 11 IP address pada server backup

Pengujian dilanjutkan dengan melakukan tes sesuai dengan skenario yaitu mematikan server utama, yang nantinya diharapkan node akan berpindah ke server *backup*. Terlebih dahulu dilakukan pengecekan pada SQL server pada server utama sudah berjalan atau belum dan terhubung ke aplikasi di DMZ nya, jika sudah terhubung maka aplikasi tidak akan menampilkan pesan error koneksi database



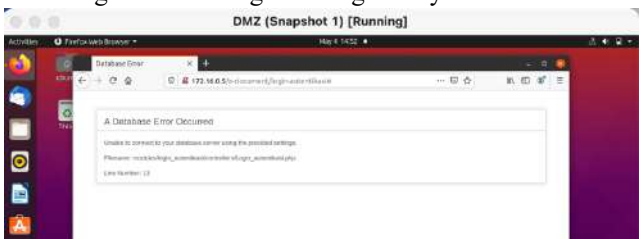
Gbr. 12 Pengecekan koneksi database dan aplikasi

Kemudian server utama dimatikan dengan melakukan stop pada GNS3 sehingga node dari server tersebut merah. Menandakan bahwa server tersebut mati atau mengalami gangguan.



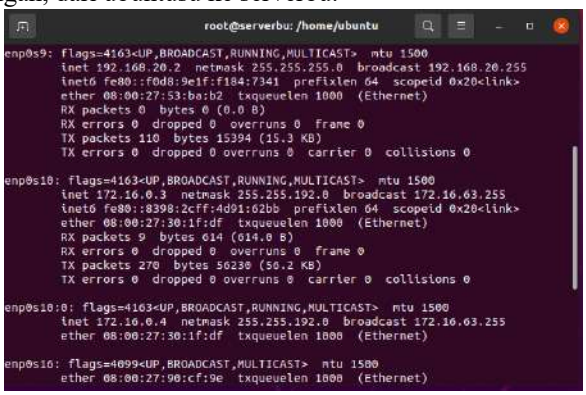
Gbr. 13 Server utama dimatikan pada GNS3

Kondisi ini menjadikan koneksi database terputus pada aplikasi sehingga muncul pesan error . Jika heartbeat bekerja maka heartbeat akan memindahkan node dari node utama ke node cadangan sesuai dengan konfigurasinya



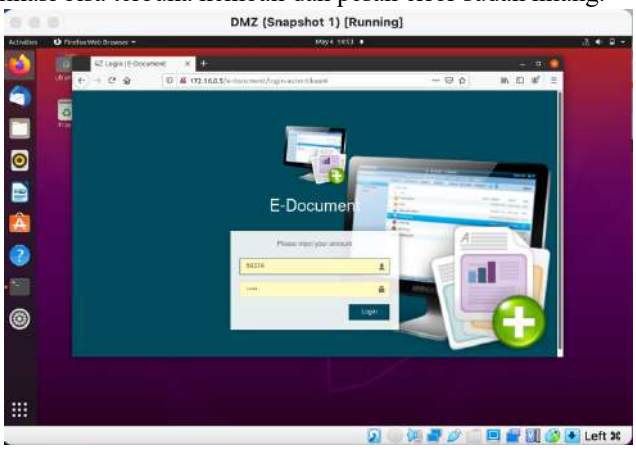
Gbr. 14 Kondisi koneksi aplikasi dan database terputus

Setelah dilakukan pengecekan maka ada perubahan pada *interfaces* serverbu (server *backup*) yakni adanya ip tambahan seperti yang ada pada server utama tadi. Bisa dipastikan heartbeat telah memindahkan node dari node utama ke cadangan, dari ubuntusu ke serverbu.



Gbr. 15 IP address server backup setelah server utama mati

Setelah node backup menyala, maka dilakukan refresh untuk melakukan pengecekan apakah database dan aplikasinya sudah terhubung atau tidak dengan web server atau DMZ. Hasilnya aplikasi bisa terbuka kembali dan pesan error sudah hilang.



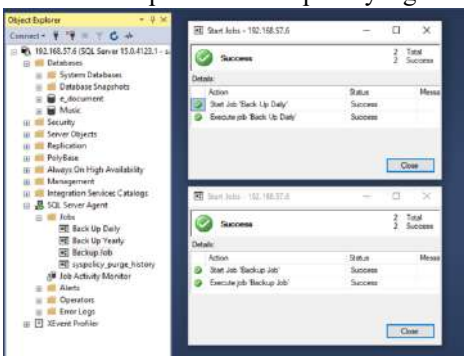
Gbr. 16 Tampilan aplikasi sesaat node server backup aktif

B. *Skenario Pemulihan Aktif*

Pada pengujian kali ini akan diujikan metode *backup* yang sudah dipersiapkan. Metode *backup* ini menggunakan jobs pada SQL agent dimana sebelumnya dikonfigurasi terlebih dahulu. SQL agent juga mengatur penjadwalan dari backup database. Jobs backup melakukan 2 macam backup yakni *backup defferential* hanya melakukan backup data yang berbeda, dan *full backup*. Pada pengujian kali ini dilakukan manual menjalankan job dengan bantuan SSMS (SQL Studio Management System). Job *backup daily* seharusnya berjalan setiap pukul 12.00 setiap harinya, dengan *backup job* berjalan pada pukul 01.00. Memberikan jeda pada masing-masing *job* untuk menghindari *load* server yang berlebih dan juga *error* saat melakukan *backup* secara otomatis.

Berikut ini hasil menjalankan job secara manual untuk kedua job backup pada sql agent. Keduanya sukses dijalankan

ditandai dengan indikator sukses dan terbentuk backup file dengan ektensi .bak pada folder yang sudah ditentukan. Untuk ip pada saat remote database menggunakan ip lokal dikarenakan ini simulasi pada satu komputer yang sama.



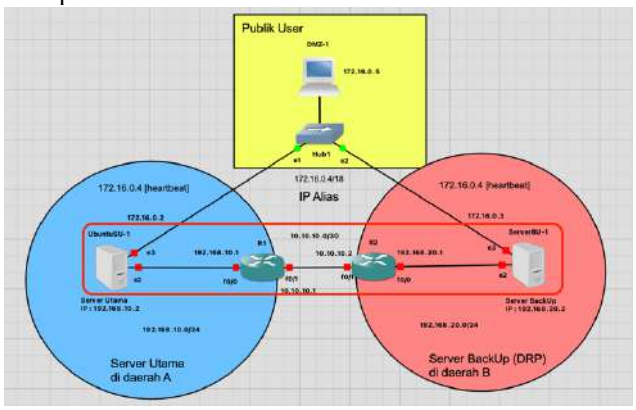
Gbr. 17 Proses manual menjalankan job backup

Berikut ini hasil dari menjalankan job secara manual. Pada folder *backup* terbentuk beberapa database *backup*. Database *backup* ini lah yang nanti dipindah ke server *backup* untuk dipulihkan dan dilakukan replikasi server.

Gbr. 18 Hasil backup pada server utama

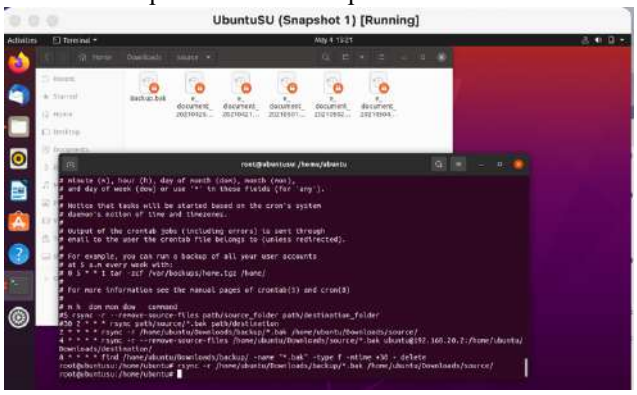
C. *Sinkronisasi data antar server*

Proses sinkronisasi ini menjadi kunci dari data yang akan direplikasi antara server. Pada proses ini data pada server utama harus bisa disinkronisasikan dengan server backup sehinga tidak ada data yang hilang. Peran manajemen waktu sangat penting antara penjadwalan backup dan penjadwalan sinkronisasi antar server. Peran sinkronisasi ini dipegang oleh aplikasi rsync dan penjadwalannya oleh crontab. Berikut ini kondisi pada simulasi.



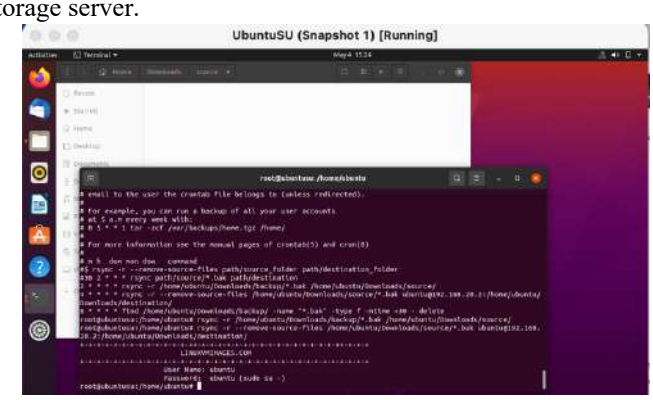
Gbr. 19 Kondisi simulasi saat sinkronisasi

Pada kondisi simulasi tersebut terdapat 2 router, router ini lah yang menghubungkan antara 2 server yang diasumsikan berada pada 2 tempat yang berbeda. Proses sinkronisasi ini seharusnya berjalan dengan otomatis namun pada pengujian kali ini akan dijalankan secara manual. Perintah rsync yang pertama akan dijalankan adalah memindahkan hasil backup dari folder backup ke folder source pada server utama.



Gbr. 20 Hasil rsync dari folder backup ke source

Selanjutnya proses sinkronisasi dengan memindahkan file dari source ke destination pada server backup serta menghapus file yang sudah dipindahkan. Penghapusan file ini untuk menghindari storage issue dimana hasil backup akan memenuhi storage server.



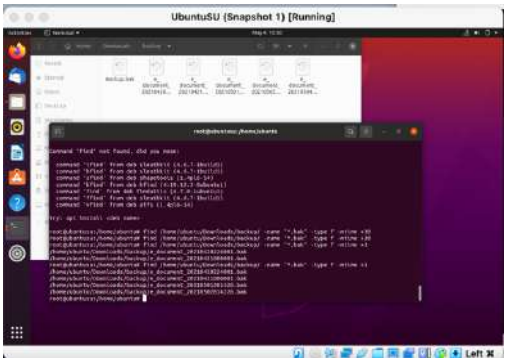
Gbr. 21 Proses pemindahan dari server uatam ke server backup

Pada proses pemindahan antar server ini selalu meminta user dan password untuk mengakses folder remote sehingga SSH disini diperlukan agar nanti pada saat melakukan pemindahan backup bisa secara otomatis tanpa melakukan pengetikan password saat crontab berjalan. Berikut ini hasil dari pemindahan file pada serverbu, seluruh file backup dari ubuntusu sebagai server utama berhasil dipindahkan.

Gbr. 22 Hasil pemindahan file backup pada server backup

Perintah pada sinkronisasi dari rsync yang terakhir adalah menghapus file backup. Penghapusan ini bertujuan untuk

mengurangi beban penyimpan server karena setiap hari melakukan backup database. Pada perintah rsync ini menghapus lebih dari 30 hari usia backup file, untuk ketentuan penghapusan ini tersendiri tergantung pada kebijakan suatu perusahaan selain itu penamaan pada file backup juga menamai dengan tanggal pada setiap bulan sehingga otomatis akan digantikan.



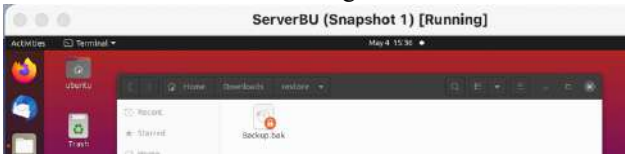
Gbr. 23 Tampilan file yang akan dihapus

D. *Pemulihan Database*

Proses terakhir dari rancangan DRP ini adalah melakukan replikasi server dengan melakukan *restore* database dari server utama ke server backup. Setelah proses sinkronisasi dilakukan maka database dari server utama akan dilakukan *restore* untuk penyamaan data. Sebelumnya hanya file backup.bak saja yang akan digunakan sehingga file tersebut dipindahkan dari folder destination ke folder restore dengan perintah rsync.

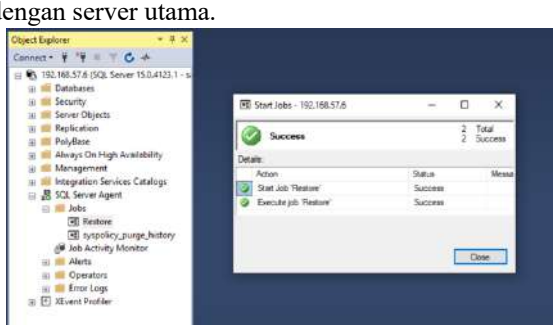
Gbr. 24 File backup pada folder destination server backup

Setelah file backup.bak sudah ada di *folder restore* maka *jobs restore* sql agent akan berjalan sehingga proses *restore database* terbaru akan berjalan dan database serverbu akan sama dengan server utama (ubuntusu). Proses *restore* ini juga seharusnya berjalan otomatis namun untuk pengujian kali ini akan dilakukan secara manual dengan bantuan SSMS.



Gbr. 25 File backup pada folder restore server backup

Hasil menjalankan job *restore* berhasil ditandai dengan indikator sukses , sehingga database pada server *backup* sudah sama dengan server utama.



Gbr. 26 Proses job restore database

Setelah melakukan *restore* pada database selanjutnya dilakukan penghapusan file backup untuk menghindari *storage issue* pada server *backup* juga, dengan ketentuan file backup lebih dari 30 hari. Semua proses DRP telah dijalankan sehingga replikasi server dan *fileover* berhasil dilakukan secara manual. Selanjutnya akan dilakukan pengujian secara otomatis untuk mengethaui waktu yang diperlukan dalam DRP berjalan.

E. *Pengujian DRP secara otomatis*

Proses terakhir dari rancangan DRP melakukan beberapa pengujian untuk mengetahui waktu yang diperlukan dalam pemulihan saat *downtime* untuk *failover* serta pengecekan data pada *database* baik server utama maupun server *backup* secara otomatis berdasarkan *schedule*. Proses pengujian diawali dengan mematikan server utama untuk mengetahui failover dari heartbeat dengan server backup, melakukan pengecekan pada database sebagai contoh dengan menggunakan database e-document pada tabel user, melakukan input manual dengan beberapa *list* data entri lalu melakukan pengecekan hasil data awal dan akhir pada masing – masing server, selanjutnya schedule backup dari differential backup diubah untuk mengetahui pengaruh dari data *loss* saat server utama dalam kondisi mati/ *down.* Kondisi crontab untuk pemindahan data dari server utama ke server backup diubah menjadi setiap 4 menit sekali, sedangkan untuk schedule backup differential pada database diubah menjadi setiap 15 menit, artinya setiap 15 menit akan dilakukan *differential backup* dalam 1 jam artinya akan ada 4 kali proses *backup differential*, sedangkan untuk full backup diubah menjadi setiap 1 jam sekali. Pada server *backup* untuk *schedule restore* diubah menjadi 10 menit sekali, artinya dalam 1 jam akan ada proses *restore* sebanyak 6 kali karena setiap 10 menit sekali proses *restore* dilakukan oleh server. Proses terakhir dari pemindahan node heartbeat dengan mengembalikan node dari server backup ke server utama sehingga nantinya saat restore dilakukan link server akan secara otomatis berpindah.

Berikut ini hasil pengujian DRP dengan waktu pada setiap kondisi.

TABEL IIIII
HASIL PENGUJIAN DRP

Waktu	Server Utama			Server Backup			DMZ	
	Down	Up	Data	Down	Up	Data	Down	Up
12.00	-	x	22	-	x	22	-	x
12.01	x	-	22	-	x	22	x	-
12.02	x	-	22	-	x	22	-	x
14.15	-	x	22	-	x	22	-	x
14.18	-	x	24	-	x	22	-	x
14.30	-	x	24	-	x	22	-	x
14.42	-	x	24	-	x	24	-	x
14.50	-	x	29	-	x	24	-	x
15.00	-	x	29	-	x	24	-	x
15.12	-	x	29	-	x	29	-	x
15.18	-	x	39	-	x	29	-	x
15.30	-	x	39	-	x	39	-	x
15.43	-	x	39	-	x	39	-	x
16.34	x	-	39	-	x	39	-	x
16.35	-	x	39	x	-	39	x	-
16.36	-	x	39	x	-	39	-	x

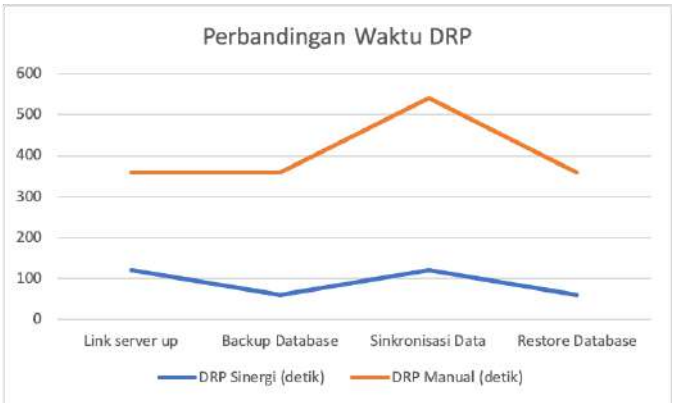
Pada tabel diatas dapat dilihat bahwa proses failover dengan heartbeat membutuhkan waktu 2 menit untuk dapat membuat server kembali normal diakses, sedangkan proses restore pada server bergantung pada waktu pemindahan data dari server utama dan proses backup pada server utama, sehingga walaupun server utama telah melakukan perubahan data, perubahan data tersebut mengalami waktu tertunda sekitar 10-20 menit setiap satu siklus DRP, hal ini menjadikan faktor penentu jika nantinya database server utama tidak dapat diakses maka kehilangan data 20 menit yang lalu tidak dapat dilakukan proses restore pada server backup.

Sebagai pembanding apakah DRP dengan sinergi replikasi dan failover ekeftif dan benar mereduksi downtime database server maka dibandingkan dengan DRP yang dilakukan dengan manual proses, perbandingan saat ini menggunakan database sama yaitu e_document dengan size 3.9 MB dengan tahapan proses yang sama dengan DRP yang sudah di rancang. Proses perbandingan meliputi proses backup linkover , backup database, sinkronisasi data, dan proses restore databasenya. Waktu ditambilkan dengan detik.

TABEL IVV
HASIL PERBANDINGAN DRP

Proses	DRP Sinergi (detik)	DRP Manual (detik)
Link server up	120	240
Backup Database	60	300
Sinkronisasi Data	120	420
Restore Database	60	300

Pada tabel perbandingan DRP diatas terlihat perbedaan waktu yang diperlukan pada masing – masing DRP, Waktu DRP sinergi dengan sistem replikasi dan failover lebih unggul dengan adanya selisih beberapa detik dengan DRP manual , bahkan ada yang memerlukan waktu lebih lama untuk DRP manual pada proses sinkronisasi data antar database server. Untuk melihat perbandingan lebih detail, berikut ini grafik perbandingan waktu antar DRP.



Gbr. 27 Grafik Perbandingan waktu DRP

Pada gambar grafik tersebut terlihat perbedaan antara DRP sinergi dengan DRP manual , beberapa proses bisa direduksi waktunya dengan signifikan menjadi bukti bahwa DRP sinergi antara sistem replikasi dan sistem failover efektif bila diterapkan untuk metode DRP bila dibandingkan dengan DRP yang dilakukan dengan manual proses.

IV. KESIMPULAN

Berdasarkan hasil dari pengujian yang telah dilakukan, maka dapat ditarik kesimpulan dalam hal failover sistem dan replikasi server untuk DRP sebagai berikut :

1. Heartbeat dapat digunakan untuk *failover* sistem menggunakan IP alias bergantung pada node yang sudah dikonfigurasikan. Jika node yang digunakan 2 menggunakan versi 1 sedangkan bila lebih dari 2 maka heartbeat versi 2 yang digunakan.
2. Replikasi server berhasil dilakukan relatif cepat dengan sistem backup full dan defferential yang merupakan kombinasi jobs backup pada sql agent dan rsync serta crontab *backup* dan *restore* database.
3. Manajemen penyimpan server tetap terjaga dengan adanya crontab untuk menghapus file *backup* secara berkala sehingga file *backup* tidak akan memenuhi server
4. Sinergi dari sistem failover heartbeat dan replikasi server mempercepat server untuk kembali *standby*, waktu rata - rata untuk *link* kembali dapat diakses 2 menit (120 detik) sedangkan untuk replikasi dan restore database membutuhkan waktu 20 menit (1200 detik).
5. Perbadingan dengan waktu downtime antara DRP sinergi dengan DRP manual menunjukan DRP sinergi mereduksi waktu downtime yang diperlukan dan efektid untuk diterapkan sebagai metode DRP.

V. SARAN

Pengujian suatu design DRP akan lebih baik dilakukan dilingkungan server asli bukan melalui simulasi bila tersedia. Pada kenyataanya akan banyak variabel diluar pengujian yang muncul selain *error* konfigurasi sehingga bisa digunakan untuk pengembangan design DRP kedepannya pada suatu perusahaan atau instansi.

UCAPAN TERIMA KASIH

Terima kasih kepada Tuhan Yang Maha Esa , orang tua, keluarga, teman – teman yang sudah mendukung saya dalam mengerjakan skripsi ini . Terima kasih juga kepada seluruh civitas akademia jurusan Teknik Informatika Universitas Negeri Surabaya, dan teman - teman tahun ajaran 2016, 2017, 2018, 2019. Semoga penelitian ini dapat digunakan untuk referensi kedepannya.

REFERENSI

[1] A. Supriyanto, I. Aknuranda, W. Hayuhardhika, and N. Putra, "Penyusunan Disaster Recovery Plan (DRP) berdasarkan Framework NIST SP 800-34 (Studi Kasus : Departemen Teknologi Informasi PT Pupuk Kalimantan Timur)," vol. 3, no. 8, pp. 8212–8219, 2019.

[2] L. Nurtanzila, "Penerapan Disaster Recovery and Contigency Planning pada Perlindungan Arsip Vital di BPN DIY," Dipl. J. Kearsipan Terap., vol. 1, no. 2, p. 82, 2018, doi: 10.22146/diplomatika.32123.

[3] R. Suryana and Y. Andrian, "Implemantasi Disaster Recovery Data Centre (Drdc) Lapan Bandung Dengan Sistem Hot Standby," Maj. Sains dan Teknol. …, pp. 1–12, 2017, [Online]. Available: https://majalah.lapan.go.id/index.php/mtsd/article/view/193.

[4] Muhaemin, "Mengembangkan Busines Continuity Planning (Bcp) dengan Pendekatan Kuantitatif Studi Kasus: Siak –Ditjen Adminduk Kemendagri," Justit Umj, vol. 9, no. 1, pp. 1–11, 2018.

[5] E. Risky Sasangka, "Simulasi Sistem Failover Komputer Clustering Menggunakan Hyper-V Pada Windows Server 2012 R2," Jurnal Manajemen Informatika, vol. 6, no. 1. 2016.

[6] M. Mulyanto and A. Ashari, "Implementasi Highly Available Website Dengan Distributed Replicated Block Device," IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 10, no. 2, p. 149, 2016, doi: 10.22146/ijccs.15528.

[7] M. I. Wahyuddin and M. Jejen, "Perancangan dan Implementasi High Availability Cluster Web Server Berbasis DBRD dan Heartbeat," pp. 205–211, 2016, doi: 10.5614/sniko.2015.30.