

Perancangan Sistem Replikasi Dan Sistem *Backup* Database *Postgresql* Menggunakan *Repmgr* Dan *Barman*

Yosef Jeffri Silvanus Nahak^{1*}, Hindriyanto Dwi Purnomo²

Teknik Informatika, Universitas Kristen Satya Wacana, Salatiga, Indonesia

*e-mail *Corresponding Author*: 672019055@student.uksw.edu

Abstract

A database is an object consisting of a set of data that is important in terms of management. If there is a disturbance to the database that makes the data from a database lost or damaged, a way is needed to recover the data and a way is needed to prevent the absence of data. The purpose of this research is to test data availability using repmgr and barman. The research begins with installing a virtual machine, installing a replication and backup system followed by testing. The result of this research is that the repmgr and barman functions on the postgresql database provide a comprehensive solution for replication and backup management in the postgresql environment. Where repmgr can simplify replication administration tasks with failover and switchover functions and ensure high availability in the postgresql database while barman performs an efficient backup system as stated in the PITR barman function.

Keywords: *Replication; Backup Recovery; Postgresql*

Abstract

Database merupakan objek yang terdiri dari sekumpulan data yang dalam hal pengelolaannya menjadi penting. Jika terjadi gangguan terhadap database yang membuat data dari suatu database hilang ataupun rusak maka diperlukan cara untuk memulihkan data serta diperlukan cara untuk mencegah dengan ketersediaan data. Tujuan penelitian ini adalah menguji avabilitas data dengan menggunakan repmgr dan barman. Penelitian diawali dengan menginstall virtual machine, instalasi sistem replikasi dan backup dilanjutkan dengan pengujian. Hasil dari penelitian ini adalah fungsi repmgr dan barman pada database postgresql memberikan solusi yang komprehensif untuk manajemen replikasi dan backup dalam environment postgresql. Dimana repmgr dapat menyederhanakan tugas administrasi replikasi dengan fungsi failover dan switchover serta memastikan high availability dalam database postgresql sedangkan barman melakukan sistem backup dengan efisien seperti tercantum dalam fungsi barman PITR.

Kata kunci: *Replikasi; pemulihan cadangan Postgresql*

1. Pendahuluan

Database merupakan sekumpulan data yang saling terkait tentang objek ataupun peristiwa sehingga dengan suatu tujuan tertentu, database perlu dirancang, dibangun dan dikumpulkan. Terdapat beberapa jenis database, salah satunya adalah Relational Database yang mengatur data berdasarkan model hubungan. Contohnya adalah PostgreSQL, yang termasuk dalam kategori database relasional[1]. Postgresql merupakan salah satu sistem manajemen basis data object-relational open source yang berkembang pesat secara global, tidak hanya menawarkan kemudahan penggunaan, tetapi juga fleksibilitas skala dan efisiensi yang tinggi[2].

Dalam perkembangannya, pengelolaan database atau basis data menjadi semakin penting, terutama ketika database yang digunakan menyimpan informasi yang krusial seperti contoh pada teknologi Big Data[3]. Jika terjadi gangguan terhadap database yang membuat data dari suatu database hilang ataupun rusak maka diperlukan cara untuk memulihkan data serta diperlukan cara untuk mencegah dengan ketersediaan data yang berhubungan langsung dengan performa dan kinerja sistem database[4].

Disinilah metode backup dan replikasi diperlukan sebagai antisipasi kehilangan data atau kerusakan data. Replikasi merupakan proses penyalinan serta maintenance database

untuk membentuk sebuah sistem terdistribusi[5]. Replikasi pada database dapat menyimpan setiap data secara berkala dikarenakan dalam replikasi mengidentifikasi dan melacak modifikasi yang terjadi pada suatu database, setelah modifikasi yang terjadi dalam database tersebut terdeteksi dan diketahui, langkah selanjutnya adalah melakukan perubahan pada database lainnya agar semuanya konsisten[6].

Backup merupakan proses menyalin file yang dapat digunakan untuk menggantikan file yang rusak atau digunakan dalam proses pemulihan data. Manajemen backup jika diterapkan dengan efektif dan tepat akan memberikan manfaat bagi setiap instansi yang terlibat, sistem backup memungkinkan penyimpanan data dengan baik dan memungkinkan pemulihan data terjadi jika dibutuhkan[7].

Repmgr merupakan suatu tool untuk mengelola dan memantau replikasi dalam database postgresql sedangkan barman merupakan tool untuk membuat dan mengelola backup database postgresql. Penggabungan tools repmgr serta barman dapat mencapai zero data loss = dan zero down-time = 0, dimana 99.99% uptime dalam setahun bisa dicapai jika, tools tersebut dikonfigurasi dengan benar dan dimonitoring dengan benar[8].

Oleh karena itu penelitian kali ini akan berfokus pada menguji avabilitas data dengan menggunakan repmgr dan barman serta mencegah adanya kerusakan serta kehilangan data pada server database serta menganalisis seberapa efektif penggunaan repmgr dan barman dalam avabilitas data pada database.

2. Tinjauan Pustaka

Beberapa penelitian sebelumnya mengenai replikasi database dan juga backup database telah dilakukan. Peneliti[9] melakukan penelitian yang berjudul "Membangun Server Dan Analisis Backup Database Postgresql Menggunakan Teknik Replication Master/Slave". Perancangan ini bertujuan untuk mengevaluasi sejauh mana data replikasi database antara server master dan server slave akurat. Tujuan utamanya adalah untuk menentukan waktu yang diperlukan agar data ter-replikasi ke server slave. Hasil analisis dari perancangan tersebut direkam dalam log yang ada di kedua server. Penelitian tersebut menunjukkan bahwa pengujian replikasi berjalan lancar tanpa adanya penundaan (waktu delay rata-rata=0).

Peneliti[10] melakukan penelitian yang berjudul "Backup Database Dengan Multi Master Replikasi Pada Kluster Server". Penelitian ini menggunakan teknik replikasi master-master dengan menggunakan MySQL galera cluster sebanyak 3 node sehingga transaksi data pada kluster database secara realtime dapat digunakan sebagai backup data. Dalam penelitian tersebut menunjukkan bahwa Teknik replikasi master-master dapat menjadi solusi untuk ketersediaan data.

Peneliti[11] melakukan penelitian yang berjudul "Simulasi Backup Dan Restore Database Repository Institusi Berbasis Aplikasi DSpace". Penelitian tersebut memiliki tujuan untuk melakukan analisis terhadap kinerja backup dan restore database pada fitur pelestarian koleksi repository institusi yang ada di aplikasi DSpace. Hasil penelitian menunjukkan bahwa penggunaan fitur pelestarian koleksi repository institusi pada aplikasi DSpace dalam melakukan backup database terjamin keamanannya, karena telah memenuhi standar keamanan informasi yang mengacu pada "CIA Triad".

Peneliti[12] melakukan penelitian yang berjudul "Implementasi Sistem Replikasi Database Postgresql Master-Slave Repmgr Dengan Auto Promote Masterdb". Penelitian tersebut memiliki beberapa tujuan salah satunya adalah untuk membuat auto promote server slave menjadi master di saat server master mengalami error. Penelitian ini menggunakan repmgr sebagai alat untuk mengelola replikasi dan failover serta menggunakan pgbouncer sebagai connection pooler. Dari penelitian tersebut terdapat hasil bahwa replikasi database berjalan dengan lancar serta sistem failover dengan repmgr dapat berjalan dengan baik.

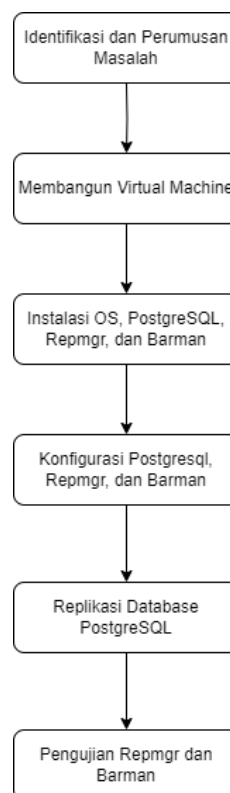
Peneliti[13] melakukan penelitian yang berjudul "Sinergi Replikasi Server Dan Sistem Failover Pada Database Server Untuk Mereduksi Downtime Disaster Recovery Planning (DRP)". Penelitian tersebut memiliki tujuan untuk mengurangi waktu downtime dari suatu server dengan menggunakan protocol heartbeat dan juga mensinergikan sistem failover dan replikasi server. Hasil penelitian tersebut menunjukkan bahwa integrasi antara sistem failover dan replikasi server berhasil mengurangi waktu downtime, sehingga mempercepat pemulihan server menjadi siap digunakan kembali. Waktu rata-rata untuk kembali dapat mengakses link adalah 2 menit, sedangkan proses replikasi dan restore database membutuhkan waktu 20 menit.

Peneliti[14] melakukan penelitian yang berjudul “Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source”. Penelitian tersebut memiliki tujuan agar tiap perubahan pada server ataupun client dapat tersimpan secara langsung di seluruh sistem. Penelitian ini menggunakan Mysql server 5.5, php, apache server serta myphp admin dan Teknik replikasi yang digunakan merupakan master-slave. Dari penelitian tersebut terdapat hasil bahwa sistem replikasi database dapat berjalan dengan lancar di Vmware dengan menggunakan sistem operasi ubuntu 14.0 dan dengan memakai sistem replikasi database dapat mengefisiensi waktu dalam suatu project.

Peneliti[15] melakukan penelitian yang berjudul ‘Implementasi Sistem Database Terdistribusi Dengan Metode Multi-Master Database Replication’. Penelitian tersebut menggunakan mekanisme replikasi multi-master. Penelitian tersebut menggunakan tools stress tester untuk melakukan pengujian kemampuan server database. Dari penelitian tersebut terdapat hasil bahwa sistem kluster database mampu mengatasi kekurangan resource komputasi pada engine database dan pada aplikasi slave mendapatkan update dari aplikasi master dengan waktu kurang dari 0.2 detik.

Adapun fokus dari penelitian ini yaitu untuk menguji avaiabilitas data dengan meminimalisir data loss dan down time menggunakan repmgr dan barman serta mencegah adanya kerusakan serta kehilangan data pada server database serta menganalisis seberapa efektif penggunaan repmgr dan barman dalam avaiabilitas data pada database.

3. Metodologi



Gambar 1 Metodologi Penelitian

Pada penelitian ini dilakukan perancangan serta melakukan implementasi sistem replikasi dan sistem backup database postgresql menggunakan repmgr dan barman. Tahapan penelitian yang dilakukan dijelaskan melalui tahapan seperti pada Gambar 1.

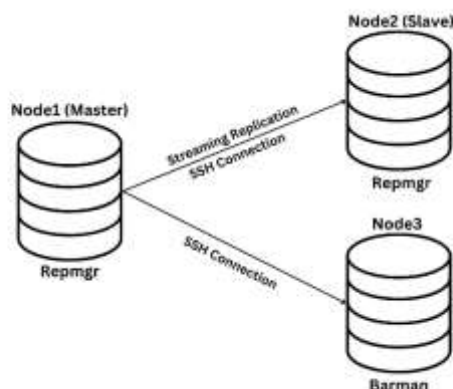
3.1 Identifikasi dan Perumusan Masalah

Pada tahap ini, peneliti melakukan identifikasi masalah dan juga perumusan masalah dengan melakukan observasi mengenai high availability dalam database beserta arsitekturnya. Peneliti juga menggunakan metode studi literatur, dimana peneliti mengumpulkan sejumlah

referensi yang berkaitan dengan high availability database melalui membaca jurnal, buku, dan penelitian serupa.

3.2 Perancangan Topologi Jaringan

Tahap ini merupakan tahap merancang jaringan yang akan digunakan untuk sistem replikasi dan backup menggunakan repmgr dan barman. Sistem ini membutuhkan 3 komponen virtual machine. Terdapat 1 node yang akan bertindak sebagai master, 1 node yang akan bertindak sebagai slave dan yang terakhir 1 node yang akan bertindak sebagai server barman. Berikut merupakan topologi yang akan digunakan :



Gambar 2 Topologi Jaringan

3.3 Membangun Virtual Machine

Peneliti membangun 3 virtual machine yang dibuat di Vmware Workstation 16.0 yang digunakan untuk sistem replikasi dan backup. Spesifikasi virtual machine yang digunakan dapat dilihat pada Tabel berikut:

Tabel 1. Spesifikasi virtual machine

Nama VM	Processor	Memory	Storage	Network
Node1	2	2	20 GB	192.168.208.194
Node2	2	2	20 GB	192.168.208.197
Node3	2	2	20 GB	192.168.208.196

3.4 Instalasi Sistem Operasi, PostgreSQL, Repmgr dan Barman

Tahap ini dilakukan instalasi sistem operasi menggunakan Redhat Enterprise Linux 8.6, Postgresql 13, Repmgr-13 serta Barman 3.2.0. Base Redhat Enterprise Linux 8.6 dapat ditemukan di website resmi redhat sedangkan instalasi postgresql, repmgr dan barman dilakukan menggunakan command-line interface dengan mengambil repositori dependensi pada website resmi serta menggunakan command `sudo yum install` untuk menginstall postgresql, repmgr dan barman.

3.5 Konfigurasi Postgresql, Repmgr dan Barman

Tahap ini dilakukan dengan mengkonfigurasi postgresql, repmgr di masing-masing node dan barman di node3. Peneliti juga mendaftarkan port firewall yang digunakan serta melakukan setup ssh-keygen di masing-masing node. Langkah-langkah ini bertujuan agar node1, node2 dan node3 dapat terhubung dengan baik sehingga dapat bekerja sama serta agar fungsi pada repmgr dan barman dapat bekerja dengan baik.

Konfigurasi postgresql yang akan dilakukan di 2 jenis file yaitu postgresql.conf dan pg_hba.conf. Dimana postgresql.conf berisi parameter-parameter sistem dari database postgresql sedangkan pg_hba.conf berisi mengenai perizinan autentikasi dan otorisasi jaringan server postgresql. Konfigurasi file postgresql.conf dilakukan dengan beberapa parameter seperti yang ada di Tabel 2.

Tabel 2. Parameter postgresql.conf

Parameter	Value
listen_addresses	''*
port	5488
shared_preload_libraries	'repmgr'
max_wal_senders	10
max_replication_slots	10
wal_level	Replica
wal_log_hints	on
archive_mode	on
archive_command	'test ! -f barman@server3:/var/lib/barman/server1/incoming/%f && rsync -a %p barman@server3:/var/lib/barman/server1/incoming/%f'

Dilanjutkan dengan memasukan ip address server yang akan digunakan untuk replikasi dalam tahap ini adalah memasukan pada file pg_hba.conf. Terdapat beberapa parameter dalam 1 line yang dimasukan seperti host, database, user, ip address, serta jenis enkripsi yang dibutuhkan, dalam hal ini peneliti menggunakan trust.

Tabel 3. Parameter replikasi pg_hba.conf

Type	Database	User	Address	Method
local	replication	all		trust
host	replication	all	127.0.0.1/32	trust
host	replication	all	:::1/128	trust
host	replication	repmgr	192.168.208.0/24	trust
host	repmgr	repmgr	192.168.208.0/24	trust
host	all	all	192.168.208.0/24	trust
host	all	postgres	192.168.208.0/24	trust
host	all	barman	192.168.208.0/24	trust
host	replication	barman	192.168.208.0/24	trust

Pada Repmgr, konfigurasi dilakukan di file repmgr.conf di ketiga node dengan mengubah node_id, node_name, conninfo, data_directory, use_replication, pg_bindir, failover, promote_command, follow_command, service_start_command, service_stop_command, service_restart_command, service_reload_command. Dilanjutkan dengan mendaftarkan server node1 menjadi master pada repmgr.

Tabel 4. Parameter repmgr.conf

Parameter	Value
node_id	1
node_name	Node1
Conninfo	'host=node1 user=repmgr dbname=repmgr port=5488 connect_timeout=5'
connect_timeout	10
data_directory	'/var/lib/pgsql/13/data/'
use_replication_slots	replica
pg_bindir	on
failover	on
promote_command	'/usr/pgsql-13/bin/repmgr -f /etc/repmgr/13/repmgr.conf standby promote -- log-to-file'
follow_command	'/usr/pgsql-13/bin/repmgr -f /etc/repmgr/13/repmgr.conf standby follow -- log-to-file'
service_start_command	'sudo systemctl start postgresql-13'
service_stop_command	'sudo systemctl stop postgresql-13'
service_restart_command	'sudo systemctl restart postgresql-13'
service_reload_command	'sudo systemctl reload postgresql-13'

Pada Barman, konfigurasi dilakukan di 2 jenis file yaitu file global barman (barman.conf) dan juga file detail server barman. Pada file global barman, terdapat beberapa parameter yang diisi ataupun diubah seperti compression, parallel_jobs, check_timeout, immediate_checkpoint, basebackup_retry_times, retention_policy, reuse_backup. Untuk file detail server barman, terdapat beberapa parameter yang diisi ataupun diubah seperti description, ssh_command, conninfo, backup_options, backup_method, archiver.

Tabel 5. Parameter barman.conf

Parameter	Value
compression	gzip
parallel_jobs	2
check_timeout	30
immediate_checkpoint	true
basebackup_retry_times	3
retention_policy	RECOVERY WINDOW OF 1 WEEKS
reuse_backup	link

Tabel 6. Parameter detail server barman

Parameter	Value
description	server1
ssh_command	ssh postgres@server1
conninfo	host=server1 user=barman dbname=postgres port=5488
backup_method	rsync
backup_options	concurrent_backup
archiver	on
path_prefix	'/usr/pgsql-13/bin'

3.6 Replikasi Database PostgreSQL

Pada tahap ini dilakukan replikasi database antar node1, node2 dengan streaming replication menggunakan command clone dari repmgr yang berfungsi untuk membackup database postgresql, mereplikasi data pada node1 ke node2. Dilanjutkan dengan pendaftaran node2 sebagai slave repmgr

ID	Name	Role	Status	Upstream	Location	Priority	Timeline	Connection string
1	node1	primary	* running		default	100	1	host=server1 user=repmgr dbname=repmgr port=5488 connect_timeout=5
2	node2	standby	running	node1	default	100	1	host=server2 user=repmgr dbname=repmgr port=5488 connect_timeout=5

Gambar 3 Hasil Replikasi

3.7 Pengujian Repmgr dan Barman

Peneliti melakukan 2 metode pengujian yang berupa pengujian fungsionalitas repmgr dan barman . Fungsi repmgr yang diuji seperti repmgr daemon, failover, switchover dan switchover. Sedangkan fungsi barman yang diuji seperti barman backup server dan barman recover menggunakan metode PITR (Point In Time Recovery).

4. Hasil dan Pembahasan

Fungsi pertama repmgr yang diuji merupakan Repmgr daemon yang merupakan suatu daemon yang dimana secara aktif memonitoring server yang berada di cluster replikasi serta melakukan beberapa tugas seperti memonitoring dan merekam performa dari replikasi serta melakukan failover dengan mendeteksi kegagalan dari server utama dan melakukan promosi ke server slave.

```

[2022-12-22 11:57:22] [NOTICE] reconnected to primary node after 10 seconds, resuming monitoring
[2022-12-22 11:57:34] [NOTICE] standby node "node2" (ID: 2) has disconnected[2022-12-22 11:59:29] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:04:29] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:09:30] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:14:30] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:16:40] [NOTICE] standby node "node2" (ID: 2) has reconnected after 1118 seconds
-bash-4.2$ [2022-12-22 12:19:31] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:24:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:29:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:34:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:39:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:44:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:49:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:54:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 12:59:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 13:04:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 13:09:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 13:14:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state
[2022-12-22 13:19:32] [INFO] monitoring primary node "node1" (ID: 1) in normal state

```

Gambar 4 Proses repmgrd

Fungsi kedua repmgr yang diuji merupakan Failover yang merupakan proses otomatis pengangkatan server slave menjadi master, dimana hal ini didorong karena adanya masalah server seperti server master yang down.

```

-bash-4.2$ /usr/pgsql-12/bin/repmgr cluster show

```

ID	Name	Role	Status	Upstream	Location	Priority	Timeline	Connection string
1	node1	primary	▲ running		default	100	19	host=node1 user=repmgr dbname=repmgr port=5488 connect_timeout=5
2	node2	standby	running		default	100	19	host=node2 user=repmgr dbname=repmgr port=5488 connect_timeout=5

Gambar 5 Cluster show

Sebelum melakukan failover peneliti melakukan input data pada database postgres untuk menguji apakah data yang tereplikasi setelah failover terjadi masih terinput dengan baik. Peneliti membuat database, Tabel dengan nama testing serta melakukan input data menggunakan generate_series sebanyak 100000 data.

```

testing=# \l

```

List of databases						
Name	Owner	Encoding	Collate	Ctype	Access privileges	
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		
repmgr	repmgr	UTF8	en_US.UTF-8	en_US.UTF-8		
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		=c/postgres
+						
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		postgres=CTc/postgres =c/postgres
+						
testing	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		postgres=CTc/postgres
(5 rows)						

```

testing=# select count(*) from testing;
count
-----
100000
(1 row)

testing=#

```

Gambar 6 Database testing

Proses failover disini akan terjadi ketika server utama mati dan saat server utama mati, tools repmgrd akan memberikan pemberitahuan warning dimana repmgr akan menunggu selama 6 x 10 detik untuk server utama hidup kembali, jika setelah 6 x 10 detik server utama tidak menyala maka proses failover akan terjadi dimana server slave akan dipromosikan menjadi server master. Disini peneliti mensetting repmgr untuk hanya menunggu selama 2 x 10 detik. Proses failover berlaku jika repmgr daemon telah berjalan.

```

[2022-12-28 14:13:02] [NOTICE]
FATAL: terminating connection due to administrator command
server closed the connection unexpectedly
This probably means the server terminated abnormally
before or while processing the request.

[2022-12-28 14:13:02] [INFO] attempting to reconnect to node "node2" [ID: 1][2022-12-28 14:13:01] [ERROR] connection to database failed
[2022-12-28 14:13:02] [DETAIL]
FATAL: the database system is shutting down
FATAL: the database system is shutting down

[2022-12-28 14:13:03] [DETAIL] attempted to connect using:
user=repmgr dbname=repmgr host=node1 port=5488 fallback_application_name=repmgr options=-csearch_path=
[2022-12-28 14:13:03] [WARNING] reconnection to node "node1" [ID: 2] failed
[2022-12-28 14:13:03] [WARNING] unable to reconnect to local node
[2022-12-28 14:13:03] [INFO] checking state of node 1, 1 of 2 attempts
[2022-12-28 14:13:03] [WARNING] unable to ping "test@repmgr connect_timeout=6 dbname=repmgr host=node1 port=5488 fallback_application_name=repmgr"
[2022-12-28 14:13:03] [DETAIL] ping() returned "PGIDM,BC,ECT"
[2022-12-28 14:13:03] [INFO] sleeping 18 seconds until next reconnection attempt
[2022-12-28 14:13:03] [INFO] checking state of node 1, 2 of 2 attempts
[2022-12-28 14:13:03] [WARNING] unable to ping "test@repmgr connect_timeout=6 dbname=repmgr host=node1 port=5488 fallback_application_name=repmgr"
[2022-12-28 14:13:03] [DETAIL] ping() returned "PGIDM,BC,ECT"
[2022-12-28 14:13:03] [WARNING] unable to reconnect to node 1 after 2 attempts
[2022-12-28 14:13:03] [NOTICE] unable to connect to local node, falling back to degraded monitoring
[2022-12-28 14:13:03] [WARNING] unable to ping "test@node1 user=repmgr dbname=repmgr port=5488 connect_timeout=5"
[2022-12-28 14:13:03] [DETAIL] ping() returned "PGIDM,BC,ECT"
[2022-12-28 14:13:03] [ERROR] unable to determine if server is in recovery
[2022-12-28 14:13:03] [DETAIL] query text is:
SELECT pg_catalog.pg_is_in_recovery()
[2022-12-28 14:13:03] [WARNING] unable to determine node recovery status
[2022-12-28 14:13:03] [WARNING] unable to ping "test@node1 user=repmgr dbname=repmgr port=5488 connect_timeout=5"
[2022-12-28 14:13:03] [DETAIL] ping() returned "PGIDM,BC,ECT"
[2022-12-28 14:13:03] [DETAIL] connection to node "node1" [ID: 3] lost
[2022-12-28 14:13:03] [DETAIL]
connection pointer is NULL.

[2022-12-28 14:13:03] [INFO] attempting to reconnect to node "node1" [ID: 1][2022-12-28 14:13:03] [ERROR] connection to database failed
[2022-12-28 14:13:03] [DETAIL]
FATAL: the database system is shutting down
FATAL: the database system is shutting down

```

Gambar 7 Warning repmgrd

Setelah proses failover maka node1 yang awalnya master akan menjadi failed dan node2 yang awalnya slave telah menjadi master melalui proses failover. Disaat node1 failed, peneliti menambahkan data sebanyak 20000 pada node2 untuk melakukan tes availability data ketika node1 sudah tersambung kembali.

Dilanjutkan dengan peneliti melakukan pengembalian status node1 menjadi running kembali dengan menggunakan *node rejoin* menggunakan command `/usr/pgsql-12/bin/repmgr node rejoin -h 192.168.208.197 -d repmgr -U repmgr -p 5488`. Maka node1 telah berjalan kembali dengan role standby.

```

~bash-4.2$ /usr/pgsql-12/bin/repmgr cluster show
ID | Name | Role | Status | Upstream | Location | Priority | Timeline | Connection string
---|---|---|---|---|---|---|---|---
1 | node1 | standby | running | node2 | default | 100 | 20 | host=node1 user=repmgr dbname=repmgr port=5488 connect_timeout=5
2 | node2 | primary | * running | | default | 100 | 20 | host=node2 user=repmgr dbname=repmgr port=5488 connect_timeout=5
~bash-4.2$

```

Gambar 8 Node rejoin berhasil

Setelah dilakukan failover akan dicek apakah database testing yang sudah ditambahkan pada saat failover terjadi masuk ke node1. Dan data yang masuk valid sebanyak 120000 data. Sehingga high availability data pada repmgr teruji.

```

testing=# select count(*) from testing;
count
-----
120000
(1 row)

```

Gambar 9 Data input 120000

Fungsi ketiga repmgr merupakan switchover yang merupakan proses pengangkatan server slave dengan sengaja menjadi master, hal ini dilakukan biasanya pada saat ingin dilakukan perubahan dari sisi master seperti penambahan storage atau lainnya. Switchover menggunakan command seperti ini `/usr/pgsql-13/bin/repmgr -f /etc/repmgr/13/repmgr.conf standby switchover`

```

[postgres@node1 ~]$ /usr/pgsql-13/bin/repmgr -f /etc/repmgr/13/fix/repmgr.conf standby switchover
NOTICE: executing switchover on node "node1" (ID: 1)
WARNING: number of pending archive files on demotion candidate "node2" exceeds the warning threshold
DETAIL: 26 pending archive files (warning threshold: 16)
HINT: PostgreSQL will not shut down until all files are archived
NOTICE: attempting to pause repmgrd on 2 nodes
NOTICE: local node "node1" (ID: 1) will be promoted to primary; current primary "node2" (ID: 2) will be demoted to standby
NOTICE: stopping current primary node "node2" (ID: 2)
NOTICE: issuing CHECKPOINT on node "node2" (ID: 2)
DETAIL: executing server command "sudo systemctl stop postgresql-13"
INFO: checking for primary shutdown; 1 of 60 attempts ("shutdown_check_timeout")
NOTICE: current primary has been cleanly shut down at location 0/17000028
NOTICE: promoting standby to primary
DETAIL: promoting server "node1" (ID: 1) using pg_promote()
NOTICE: waiting up to 60 seconds (parameter "promote_check_timeout") for promotion to complete
NOTICE: STANDBY PROMOTE successful
DETAIL: server "node1" (ID: 1) was successfully promoted to primary
NOTICE: node "node1" (ID: 1) promoted to primary, node "node2" (ID: 2) demoted to standby
NOTICE: switchover was successful
DETAIL: node "node1" is now primary and node "node2" is attached as standby
NOTICE: STANDBY SWITCHOVER has completed successfully
[postgres@node1 ~]$

```

Gambar 10 Proses switchover

Dimana dalam proses ini akan terjadi pertukaran role antara node1 dan node2. Node1 yang merupakan server slave akan menjadi server master dan sebaliknya yaitu node2 yang merupakan server master akan menjadi server slave. Perintah switch over dijalankan di server slave.

Pengujian dilanjutkan dengan tool barman dimana fungsi pertama yang diuji merupakan barman backup menggunakan command `barman backup Master1 (node1)`


```
[barman@barman1 ~]$ barman backup Master1
Starting backup using rsync-concurrent method for server Master1 in /var/lib
/barman/Master1/base/20220922T203841
Backup start at LSN: 0/C000060 (00000001000000000000000C, 00000060)
Starting backup copy via rsync/SSH for 20220922T203841 (2 jobs)
Copy done (time: less than one second)
Asking PostgreSQL server to finalize the backup.
Backup size: 32.4 MiB. Actual size on disk: 24.9 MiB (-99.93% deduplication
ratio).
Backup end at LSN: 0/C000138 (00000001000000000000000C, 00000138)
Backup completed (start time: 2022-09-22 20:38:42.090204, elapsed time: 4 se
conds)
Processing xlog segments from file archival for Master1
00000001000000000000000000000000
00000001000000000000000000000000
00000001000000000000000000000000.00000060.backup
[barman@barman1 ~]$
```

Gambar 11 Barman backup

Fungsi kedua barman yang diuji merupakan Point In Time Recovery yang akan berperan penting dalam meminimalisir data loss dikarenakan PITR barman ini dapat melakukan recovery server berdasarkan timestamp, transaction ID serta restore label. Dalam pengujian kali ini peneliti menggunakan opsi timestamp yaitu target time

```
[barman@barman1 ~]$ barman recover --remote-ssh-command "ssh postgres@master
2" Master1 20221005T192022 /var/lib/pgsql/12/data --target-time "2022-10-05
19:21:30"
Starting remote restore for server Master1 using backup 20221005T192022
Destination directory: /var/lib/pgsql/12/data
Remote command: ssh postgres@master2
Doing PITR. Recovery target time: '2022-10-05 19:21:30-07:00'
Copying the base backup.
Copying required WAL segments.
Generating recovery configuration
Identify dangerous settings in destination directory.

IMPORTANT
These settings have been modified to prevent data losses

postgresql.conf line 234: archive_command = false

Recovery completed (start time: 2022-10-05 19:47:32.257132-07:00, elapsed ti
me: 10 seconds)
Your PostgreSQL server has been successfully prepared for recovery!
```

Gambar 12 Barman PITR

Tabel 7. Pengujian Fungsi Repmgr

Fungsi Uji	Cara Pengujian	Hasil yang diharapkan	Hasil Pengujian
Repmgrd	Menggunakan command repmgrd	Tampil daemon monitoring	Valid
Failover	Mematikan server master	Server slave menjadi master secara otomatis	Valid
High Availability Data	Menambahkan data saat failover terjadi di node2	Data tetap terinput di node1	Valid
Switchover	Menggunakan command standby switchover	Server slave bertukar peran dengan server master	Valid

Tabel 8. Pengujian Fungsi Barman

Fungsi Uji	Cara Pengujian	Hasil yang diharapkan	Hasil Pengujian
Full Backup	Menggunakan command barman backup	Backup database berjalan dengan lancar tanpa error & corrupt	Valid
Recover with PITR	Menggunakan command barman recover	Recovery database berjalan lancar tanpa error	Valid

Dari hasil pengujian secara fungsional pada sistem replikasi dan sistem backup menggunakan barman secara keseluruhan fungsionalitas berjalan dengan baik dan lancar. Peneliti tidak melakukan uji testing performa seperti benchmarking ataupun uji query respon time dikarenakan repmgr dan barman merupakan tools manajemen replikasi dan backup yang berfungsi untuk memaksimalkan high availability dalam database postgresql.

Konsep penelitian ini merupakan penelitian yang sejalan dengan penelitian sebelumnya Erwin Asriyar, Teten Sutendi [12] yang dimana telah melakukan implementasi sistem replikasi master-slave menggunakan repmgr. Akan tetapi pada penelitian ini lebih memaksimalkan fitur pada repmgr seperti repmgrd dan switchover. Konsep penelitian ini juga sejalan dengan penelitian oleh Ahmad Heryanto [10] yang dimana telah melakukan backup database menggunakan sistem replikasi multi master. Akan tetapi pada penelitian ini menggunakan replikasi master slave dengan bantuan repmgr dan barman.

5. Simpulan

Replikasi dan backup sangatlah diperlukan sebagai antisipasi kehilangan data atau kerusakan data. Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa fungsi repmgr dan barman pada database postgresql memberikan solusi yang komprehensif untuk manajemen replikasi dan backup dalam environment postgresql. Dimana repmgr dapat menyederhanakan tugas administrasi replikasi dengan fungsi failover dan switchover serta memastikan high availability dalam database postgresql sedangkan barman melakukan sistem backup dengan efisien seperti tercantum dalam fungsi barman PITR. Fitur otomatisasi dan manajemen yang disediakan oleh repmgr dan barman dapat menjaga ketersediaan tinggi dan pemulihan yang cepat dalam situasi darurat. Saran pengembangan untuk penelitian selanjutnya yaitu dengan mengeksplorasi penggunaan parameter konfigurasi repmgr dan barman yang lebih optimal serta penggunaan lebih banyak cluster database serta menggunakan environment cloud untuk pengujian kinerja, kapasitas dan skalabilitas yang lebih baik.

Daftar Referensi

- [1] S. Endang, H. Wijoyo, and N. Soeharmoko, *Relational Database Management System (RDBMS)*, vol. 1. CV. Pena Persada, 2020.
- [2] L. Ferrari and E. Pirozzi, *Learn PostgreSQL: Build and manage high-performance database solutions using PostgreSQL 12 and 13*. Packt Publishing Ltd, 2020.
- [3] M. N. Y. Utomo, "Pengembangan Model Migrasi Database Relational ke NoSQL Memanfaatkan Metadata SQL," *Jurnal Teknologi ElektriKa*, vol. 17, no. 2, pp. 1–6, 2020.
- [4] B. A. Setiawan, N. H. Sutanto, G. F. Rahman, and E. Utami, "Pengamanan Backup dan Restore Basis Data dengan Penambahan Enkripsi Advanced Encryption Standard (Studi Kasus: Analisis Jabatan Bagian Organisasi Kabupaten Balangan)," *Jurnal Sistem Komputer dan Informatika*, vol. 2, no. 3, pp. 277–282, 2021.
- [5] D. Widyasto, "Penerapan Replikasi Asynchronous Pada Basis Data Terdistribusi Untuk Keamanan Data," *Teknois: Jurnal Ilmiah Teknologi Informasi dan Sains*, vol. 6, no. 1, pp. 62–74, 2019.
- [6] M. H. Darmawan, "Perancangan dan Implementasi Sistem Replikasi Database Terdistribusi Pada Fakultas Teknik Universitas Halu Oleo," *J. T. INFORMATIKA*, vol. 4, pp. 91–98, 2018.
- [7] E. Haryadi, A. Abdussomad, and R. Robi, "Implementasi Sistem Backup Data Perusahaan Sebagai Bagian dari Disaster Recovery Plan," *SAINSTECH: Jurnal Penelitian dan Pengkajian Sains dan Teknologi*, vol. 29, no. 2, pp. 6–11, 2019.
- [8] A. Menon-Sen, D. Michel, and M. Wallace, "Barman Manual," Mar. 29, 2023. <https://docs.pgbarman.org/release/3.5.0/> (accessed Apr. 14, 2023).
- [9] Z. Arifin, J. Triyono, and R. Y. Rachmawati, "MEMBANGUN SERVER DAN ANALISIS BACKUP DATABASE POSTGRESQL MENGGUNAKAN TEKNIK REPLICATION MASTER/SLAVE," *Jurnal SCRIPT*, vol. 7, no. 1, pp. 107–114, 2019.
- [10] A. Heryanto and Y. Hartati, "Backup Database Dengan Multi Master Replikasi Pada Kluster Server," *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, vol. 6, no. 1, pp. 7–12, 2020.
- [11] I. Arnomo, "SIMULASI BACKUP DAN RESTORE DATABASE REPOSITORY INSTITUSI BERBASIS APLIKASI DSPACE," *JUST IT: Jurnal Sistem Informasi*, vol. 9, no. 2, pp. 92–99, 2019.

- [12] E. Asriyar and T. Sutendi, "Implementasi Sistem Replikasi Database PostgreSQL Master-Slave REPMGR dengan Auto Promote MasterDB," *Jl-Tech*, pp. 8–29, 2019.
- [13] W. A. Yuliono and A. Prihanto, "Sinergi Replikasi Server dan Sistem Failover pada Database Server untuk Mereduksi Downtime Disaster Recovery Planing (DRP)," *Journal of Informatics and Computer Science (JINACS)*, vol. 3, no. 01, pp. 29–38, 2021.
- [14] H. Maulana, "Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source," *Info TekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 1, no. 1, pp. 32–37, 2016.
- [15] A. Heryanto and A. Albert, "Implementasi Sistem Database Terdistribusi Dengan Metode Multi-Master Database Replication," *Jurnal Media Informatika Budidarma*, vol. 3, no. 1, pp. 30–36, 2019.