

# Sistem Replikasi Basis Data berdasarkan MySQL menggunakan *Container Docker*

Ilmiyatus Sholihah<sup>1</sup>, Cahyo Darujati<sup>2</sup>

[Submission: 18-06-2022, Accepted: 28-08-2022]

**Abstract**—This study aims to implement database replication using the master-slave technique. The main focus of this research is to run data replication on the master server so that it can be copied automatically on the slave server. Master-slave replication techniques aim to prevent data loss from system failure events or device damage on the main server. The concept of master-slave replication is to copy data from the master server database to the slave server so that users do not need to manually backup data from the database. Master-slave database replication is one of the features provided by the MySQL database application in the Docker container. Still, to run the replication system, some adjustments and configurations are required for the database application to establish a replication infrastructure automatically. In this study, the replication was run on a MySQL database application with 20 multimedia data files capacity of 100MB. Testing is done by entering data into the database on the master server. The test scenario in this study was to upload data and delete data. The test results show that the data changes automatically: the data on the slave server is the same as on the master server, even though the data on the master server has gone through data uploads and deletions. And vice versa, data changes made on the master server are also directly replicated on the slave server.

**Intisari**—Penelitian ini bertujuan untuk melakukan implementasi replikasi database menggunakan teknik master-slave. Fokus utama dari penelitian ini adalah menjalankan replikasi data yang terdapat pada server master sehingga dapat tersalin secara otomatis pada server slave. Penerapan teknik replikasi master-slave bertujuan untuk mencegah kehilangan data dari kejadian sistem gagal atau kerusakan perangkat pada server utama. Konsep replikasi master-slave adalah menyalin data dari database server master ke server slave sehingga pengguna tidak perlu melakukan backup data dari database secara manual. Replikasi database master-slave merupakan salah satu fitur yang disediakan oleh aplikasi database MySQL dalam container Docker, namun untuk dapat menjalankan sistem replikasi, beberapa penyesuaian dan konfigurasi diperlukan pada aplikasi database, sehingga mampu membentuk sebuah infrastruktur replikasi secara otomatis. Dalam penelitian ini, replikasi dijalankan pada aplikasi database MySQL dengan data uji sebanyak 20 file data multimedia dengan kapasitas 100MB. Pengujian dilakukan dengan memasukkan data ke dalam database pada server master. Skenario pengujian pada penelitian ini dilakukan unggah data dan hapus data. Dari hasil pengujian

menunjukkan perubahan data secara otomatis: data yang ada pada server slave sama seperti pada server master, walaupun data pada server master telah melalui unggahan dan penghapusan data. Begitu pula sebaliknya, perubahan data yang dilakukan pada server master juga langsung terreplikasi pada server slave.

**Kata Kunci**—Replikasi Basis Data, MySQL, Docker, Container, Database.

## I. PENDAHULUAN

Era digital Industri 4.0 saat ini mendorong setiap perusahaan untuk menyimpan data secara strategis. Data dan informasi, terlebih milik perusahaan, merupakan suatu hal penting yang menjadi roda penggerak ekonomi suatu perusahaan [1]. Untuk itu, aplikasi penyimpanan data pada komputer dirancang sedemikian rupa untuk dapat menyajikan data yang cepat dan efisien [2]. Kegiatan rutin yang melibatkan data menyebabkan pihak pengelola basis data atau database harus melakukan backup data secara rutin sehingga perubahan yang terjadi dalam database terbaru setiap saat [3]. Backup dilakukan untuk mengantisipasi terjadinya perubahan dan kehilangan data.

Semakin pendek rentang waktu terjadinya antara proses dengan backup data maka peluang untuk memperoleh data yang akurat akibat terjadinya perubahan ataupun kehilangan data akan semakin besar [4]. Replikasi adalah suatu teknik untuk melakukan penyalinan dan pendistribusian data dan objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin [5]. Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet.

Pada penelitian ini, dilakukan simulasi backup database MySQL menggunakan replikasi master-slave pada dua komputer server, dan kontainer Docker. Target dari penelitian ini adalah melakukan backup secara otomatis dari server master ke server slave satu arah, data pada database yang diisi dari server master akan ada secara otomatis berada pada database server slave.

Adapun penelitian ini dilakukan karena belum adanya system yang menggunakan Replikasi Berbasis Data untuk digunakan dalam suatu perusahaan. Terlebih, penggunaannya masih terlalu rumit, sehingga membutuhkan keahlian khusus dalam bidang IT. Oleh sebab itu, penelitian ini membahas mengenai sistem replikasi data berdasarkan MySQL menggunakan Container Deck. Selain itu, bias memberikan rekomendasi kepada pembaca mengenai keefektifan penggunaan system ini di perusahaan.

## II. PENELITIAN TERDAHULU



Replikasi adalah suatu teknik untuk melakukan penyalinan dan pendistribusian data dan objek-objek *database* dari satu *database* ke *database* lain dan melaksanakan sinkronisasi antara *database* sehingga konsistensi data dapat terjamin. Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet. Replikasi dengan teknik *master-slave* digunakan untuk memecahkan masalah seperti performa, mendukung cadangan *database* yang berbeda dan solusi untuk mengurangi kegagalan dari sebuah sistem. Hal ini memungkinkan data dari satu *database server* (*master*) untuk direplikasi ke satu atau lebih *database server* lain (*slave*) [6]. Replikasi *master-slave* mewujudkan peralihan otomatis secara efektif memastikan ketersediaan *database* yang tinggi ketika *database* utama gagal [7].

Arni pada penelitiannya merancang sistem replikasi *database* yang berjalan diatas *virtual machine*. Penelitian ini mendapatkan kesimpulan bahwa replikasi *database* dapat berjalan dengan baik pada VMware Workstation dengan system operasi Ubuntu Server 14.0. Menurut Arni, sistem replikasi *database* dapat mempermudah, mempercepat dan menghemat baik dari segi proses maupun waktu dalam suatu proyek [6].

Shrestha dkk selanjutnya juga melakukan replikasi *database* pada *cloud* dengan menggunakan Galera Cluster. Pada penelitian Shrestha dkk, dihasilkan kesimpulan mengenai hasil dari pengujian *high availability*. Menurutnya, Galera Cluster mampu bekerja dengan baik pada saat *database* bekerja dengan beban tinggi yang dijalankan secara *cluster* [8].

Penelitian milik Subekti dkk. membahas tentang perancangan replikasi *database* yang berjalan pada *failover clustering*. Penelitian ini menghasilkan simpulan bahwa *replication mirror* dan dengan bantuan *failover clustering* yang telah dirancang, dapat menjadi pilihan solusi yang terbaik, karena dapat menggantikan *server* yang mati ke *server* yang lain dan *client* tetap bisa mengakses *web server* [9].

### III. METODE PENELITIAN

#### A. Database MySQL

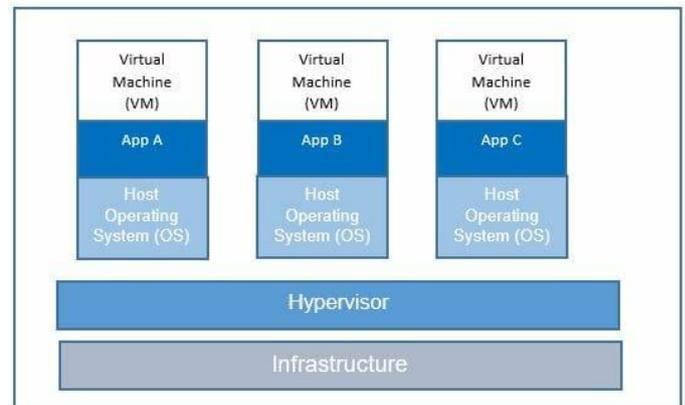
MySQL adalah sistem manajemen *database* relasional (RDBMS) berbasis SQL (*Structured Query Language*) yang bersifat *open-source* dengan model *client-server* [10]. Secara umum, MySQL adalah perangkat lunak yang berfungsi untuk membuat dan mengelola berbagai informasi yang ada pada *database* di dalam sebuah *server* dengan bahasa pemrograman SQL [11], [12].

Kelebihan MySQL dibandingkan dengan RDBMS lain adalah mudah, gratis, stabil, dan dapat diterapkan pada beberapa sistem operasi yang berbeda [13]. MySQL saat ini penggunaannya sangat populer di dunia IT, sehingga tidak diragukan lagi tentang performa serta kinerja dalam menjalankan transaksi data.

#### B. Virtual Machine

Secara singkat, *Virtual Machine* (VM) merupakan sistem operasi tambahan yang bekerja pada lingkungan server virtual layaknya sistem operasi bawaan pada komputer. Dengan kata lain, mesin virtual ini adalah perangkat yang membuat komputer di dalam komputer, berjalan sebagaimana seperti

program lainnya [14]. Ada beberapa jenis arsitektur pada *virtual machine*, diantaranya ada adalah arsitektur model *hypervisor* seperti pada Gambar 1. Arsitektur *virtual machine hypervisor* memiliki sistem operasi kedua yang berjalan diatas sistem operasi utama [15]. Aplikasi yang umum untuk *virtual machine* adalah VMware dan VirtualBox. Penggunaan *virtual machine* dapat menghemat biaya riset pada pengembangan perangkat lunak, termasuk uji coba terhadap perangkat lunak *database*.



Gambar 1: Arsitektur *Virtual Machine*

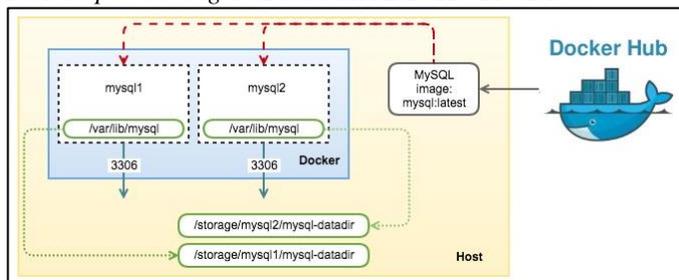
#### C. Sistem Operasi Server

Sistem operasi adalah bagian pertama yang ditempatkan pada memori komputer pada saat komputer dinyalakan (*booting*). Sistem operasi juga memiliki banyak jenis, diantaranya sistem operasi *server*. Saat ini, sistem operasi *server* yang umum digunakan adalah dari sistem operasi Linux. Terdapat dua jenis sistem operasi *server* pada Linux, yaitu Debian Server dan RedHat. Perbedaan pada kedua sistem operasi ini adalah RedHat berbayar untuk mendapatkan support dari pengembang, sedangkan Ubuntu Server bersifat gratis dan mendapat dukungan penuh dari komunitas.

Linux Ubuntu Server mempunyai lisensi *open source* serta merupakan turunan dari distro Linux Debian, sehingga mempunyai keamanan yang cukup tinggi. Seperti pada kegunaan utamanya, Ubuntu Server tidak disediakan *Graphical User Interface* (GUI). Hal itu berarti semua bentuk eksekusi program dijalankan melalui baris perintah yang disebut dengan *Command Line Interface* (CLI). Saat ini Ubuntu sudah banyak digunakan di berbagai perusahaan besar di seluruh dunia, baik versi *desktop* maupun yang *server* [16].

#### D. Docker sebagai Container

*Container* pada aplikasi saat ini sedang populer digunakan oleh para *developer*. Salah satu *container* bernama Docker menawarkan teknologi *container* pada aplikasi, yang sebelumnya biasa digunakan pada sistem operasi [17].



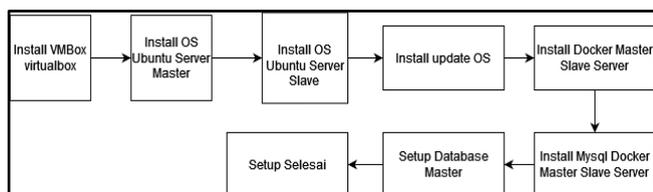
Gambar 2: Ilustrasi Container dalam Docker

Container pada aplikasi ini sangat cocok digunakan untuk desain arsitektur seperti *microservice*, karena masing-masing service memiliki lingkungan yang terisolasi dan tetap bisa berkomunikasi satu sama lain. Selain daripada itu, container menawarkan kemudahan dalam melakukan migrasi [18], yang mana memiliki konsep mirip seperti replikasi *database*.

### E. Persiapan Eksperimen

Penelitian ini memfokuskan replikasi *database* MySQL dengan teknik replikasi *master to slave* pada VirtualBox. Penelitian ini menggunakan *script* untuk mengetahui data pada aplikasi *database* sudah benar terduplikasi dengan baik atau tidak. Data yang digunakan pada penelitian ini berupa data *file* multimedia video dengan kapasitas 100MB yang akan dikirim ke dalam *database* MySQL dengan menggunakan aplikasi sederhana untuk *query insert* kedalam *database*.

Analisa terhadap duplikasi data sangat penting dalam penelitian ini, dikarenakan fokus utama pada penelitian ini melakukan replikasi dan keberhasilan *database* terduplikasi pada *server slave* menjadi parameter utama. Hasil analisa akan dikaji untuk mengetahui seberapa optimal duplikasi *database* dengan metode *master to slave*. Gambar 3 menunjukkan ilustrasi dari alur persiapan eksperimen dalam bentuk blok diagram.



Gambar 3: Blok Diagram Alur Persiapan Eksperimen

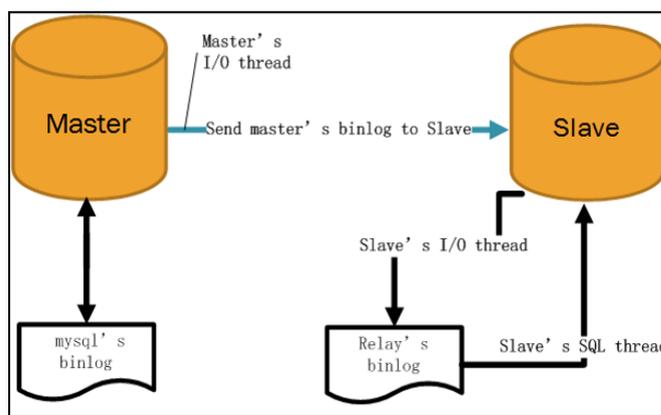
Untuk memulai penelitian, dilakukan konfigurasi pada *server* dengan melakukan instalasi sistem operasi utama pada *server* VM yang akan digunakan. Setelah dilakukan instalasi sistem operasi utama dilanjutkan dengan melakukan instalasi *library* tambahan standar *server* dan *tools monitoring* yang digunakan pada penelitian ini. Fungsi dari instalasi *library* pendukung adalah untuk membantu agar sistem operasi tidak kehilangan *library* yang dibutuhkan yang berakibat terhambatnya penelitian.

Ilmiyatus Sholihah: Sistem Replikasi Basis Data...

Setelah dilakukan instalasi *library* pada sistem operasi *server*, dilanjutkan dengan melakukan instalasi Docker. Setelah melakukan instalasi Docker, dilanjutkan dengan instalasi aplikasi *database* MySQL dengan Docker. Setelah melakukan instalasi MySQL dengan Docker, dilakukan konfigurasi melalui perintah *script* pada setingan *database* MySQL. Jika tidak ada *error*, maka proses *setup environment* telah selesai.

### F. Setup Pengujian Eksperimen

Penelitian ini menggunakan spesifikasi perangkat keras CPU Core i3 berkecepatan 2Ghz dengan arsitektur 64bit. Untuk memori RAM menggunakan kapasitas 4Gb dan media penyimpanan sebesar 20GB. Pada setiap VM, digunakan kapasitas masing-masing 10GB pada VM *master* dan VM *slave*, dengan kapasitas masing-masing 1GB. Juga ditambahkan beberapa *tools* untuk melakukan konfigurasi pada *server* yang akan berfungsi sebagai alat bantu untuk mengetahui fungsi dari replikasi *database*.



Gambar 4: Blok Diagram Alur Pengujian Eksperimen

Pada Gambar 4, *database master* dan *database slave* saling terhubung. Pada pengujian penelitian ini, data akan diunggah ke dalam *database master* dengan konfigurasi yang telah ditentukan untuk dapat menjalankan proses replikasi *master-slave*. Setelah melakukan pengisian data pada tabel pada *database master*, dilakukan pengecekan pada *database server slave* dengan melakukan *query*. Jika data yang telah diisi pada *server database master* ada pada *server database slave*, maka proses replikasi telah selesai dan berhasil.

## IV. HASIL DAN PEMBAHASAN

### A. Setup Pengujian pada Docker

Sebelum melakukan eksperimen, diperlukan konfigurasi pada infrastruktur yang akan digunakan pada penelitian. Pada pengaturan Docker MySQL, dituliskan alamat *directory* "datadir" yang berfungsi sebagai tempat untuk data tersimpan. Dalam *directory* tersebut, tersimpan *log* dari *database* dan besar kapasitas dari data yang ada pada *database*.

p-ISSN:1693 – 2951; e-ISSN: 2503-2372



```
#!/bin/bash
docker run \
-v `pwd`/conf.d:/etc/mysql/conf.d \
-v `pwd`/datadir:/var/lib/mysql \
-v `pwd`/backup:/backup \
-e MYSQL_ROOT_PASSWORD='Admin1234$' \
-e TZ=Asia/Jakarta \
-p 3306:3306 \
--restart always \
--name mysql-master \
-d mysql:5.7.22
```

Gambar 5: Alamat “datadir” dan metadata lainnya dalam Docker

Pada Gambar 5, ditunjukkan alamat dari “datadir” database replikasi. Pada “datadir”, akan ditampung data dari database, semakin banyak dan besar data yang dimasukkan, maka semakin besar juga kapasitas directory database. Setelah melakukan konfigurasi pada “datadir”, dilakukan instalasi MySQL dengan menjalankan script Docker.

Gambar 6 menunjukkan proses instalasi MySQL yang berjalan diatas Docker. Proses instalasi dilakukan secara bersamaan pada server master dan server slave guna menyingkat waktu penyiapan media pengujian. Setelah melakukan instalasi pada Docker, dilakukan pengecekan container Docker MySQL. Setelah proses instalasi Docker dan MySQL selesai, dilakukan konfigurasi replikasi pada MySQL.

```
server@servermaster:~$ cd /srv/docker/mysql-master/
server@servermaster:~/srv/docker/mysql-master$ sudo ./rundocker.sh
Unable to find image 'mysql:5.7.22' locally
5.7.22: Pulling from library/mysql
b488812b4556: Extracting [=====] | 15.6MB/22.49MB
c39554db1d7: Download complete
9931f4da3586: Downloading [=====] | 1.9739B/4.4999B
b01b69e4f9a: Download complete
a65f125f718: Download complete
2d9f8d899a2: Downloading [=====] | 7.8420B/12.899B
376912cb2afe: Download complete
7959212cb7f: Downloading [=====]
68f949d681d: Waiting
79c45464778: Waiting
2b3c2e6ace: Waiting

server@serverslave:~$ cd /srv/docker/mysql-slave/
server@serverslave:~/srv/docker/mysql-slave$ sudo ./rundocker.sh
Unable to find image 'mysql:5.7.22' locally
5.7.22: Pulling from library/mysql
b488812b4556: Downloading [=====] | 14.239B/22.49MB
c39554db1d7: Download complete
9931f4da3586: Download complete
b01b69e4f9a: Download complete
a65f125f718: Download complete
2d9f8d899a2: Downloading [=====] | 5.6140B/12.899B
376912cb2afe: Download complete
7959212cb7f: Downloading [=====]
68f949d681d: Waiting
79c45464778: Waiting
2b3c2e6ace: Waiting
```

Gambar 6: Proses Instalasi MySQL dalam Docker

**B. Konfigurasi Master-Slave pada MySQL**

Setelah melakukan pengaturan pada Docker, selanjutnya dilakukan pengaturan pada database MySQL. Terdapat dua pengaturan konfigurasi, yaitu pada server master dan server slave. Dalam pengaturan database, ada beberapa parameter yang tidak sama antara server master dan server slave.

Langkah pertama untuk konfigurasi server master yaitu pembuatan user database untuk replikasi. Fungsi dari user ini adalah sebagai penanda replika database yang dituju.

```
mysql> create user 'repl'@'%' identified by 'replicate' password expire never;
Query OK, 0 rows affected (0.02 sec)

mysql> grant reload, replication client, replication slave on *.* to 'repl'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> create user 'user'@'%' identified by 'testing' password expire never;
```

Gambar 7: Konfigurasi hak akses pengguna yang akan melakukan replikasi

```
server@servermaster:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| uploadfile |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Gambar 8: Konfigurasi database “uploadfile” pada server master

```
mysql> select * from uploadfile;
+-----+-----+
| IDuploadfile | FilePath |
+-----+-----+
| 23 | C:\Users\Admin\Documents\dataset\Capture9.PNG |
| 24 | C:\Users\Admin\Documents\dataset\Capture10.PNG |
| 25 | C:\Users\Admin\Documents\dataset\Capture11.PNG |
| 26 | C:\Users\Admin\Documents\dataset\Capture12.PNG |
| 27 | C:\Users\Admin\Documents\dataset\Capture13.PNG |
| 28 | C:\Users\Admin\Documents\dataset\CLOSER_2017_39.pdf |
| 29 | C:\Users\Admin\Documents\dataset\Design and Implementation of a MySQL Database Backup.pdf |
| 30 | C:\Users\Admin\Documents\dataset\WIKOLAY_Baychenko.pdf |
| 31 | C:\Users\Admin\Documents\dataset\ONLampMySQLRepl.pdf |
| 32 | C:\Users\Admin\Documents\dataset\Knewella - Live For The Night.mkv |
+-----+-----+
10 rows in set (0.02 sec)
```

Gambar 9: Isi dari database “uploadfile”

```
server@servermaster:~$ mysqldump --single-transaction --quick --root -p --databases uploadfile | gzip > uploadfile.sql.gz
Enter password:
Warning: A partial dump from a server that has GTIDs will by default include the GTIDs of all transactions, even those that changed suppressed parts of the database. If you
don't want to restore GTIDs, pass --set-gtid-purged=OFF. To make a complete dump, pass --all-databases --triggers --routines --events.
server@servermaster:~$
```

Gambar 10: Proses dump database “uploadfile” pada server master

```
replicate_do_db=uploadfile
query_cache_size=5M
query_cache_type=1
max_binlog_size=1G
skip_name_resolve
gtid_mode=ON
enforce_gtid_consistency
default_password_lifetime=0
max_connections=300
default_time_zone='+7:00'
character_set_server=utf8
```

Gambar 11: Konfigurasi database pada server slave

Gambar 7 adalah command untuk membuat user dan hak akses pada database master. User ini diberi konfigurasi “password expire: never”, yang berfungsi agar password tidak

perlu diganti setiap tahun. Penggantian *password* setiap tahun merupakan ciri dari MySQL versi 5.7. Setelah melakukan konfigurasi pada *user database*, dibuat sebuah *database* dengan nama “*uploadfile*”. Gambar 8 menunjukkan bahwa *database* “*uploadfile*” berhasil terbuat. Pada *database* “*uploadfile*”, juga dibuat tabel untuk menyimpan ID. Tabel ini akan secara otomatis terisi oleh data pengujian, ketika dilakukan replikasi.

Data yang akan dimasukkan ke dalam *database* “*uploadfile*” adalah gambar dan *file* dengan format PDF. Gambar 9 menunjukkan bahwa data sudah di dalam *database* “*uploadfile*”. Lalu dilakukan proses dumping pada *database* “*uploadfile*” yang akan dijalankan ke dalam *server slave*. Proses *dump* berfungsi sebagai media sinkronisasi antara *database master* dan *database slave*, sehingga kedua *database* tersebut dapat berkomunikasi jika ada perubahan di dalam *database*. Proses *dump* ditunjukkan pada Gambar 10.

*File dump* pada proses ini akan di-*restore* atau dikembalikan ke dalam *database slave* sekaligus dengan beberapa parameter untuk menjalankan replikasi. Sampai pada langkah ini, konfigurasi pada *database MySQL server master* telah selesai dan berlanjut pada konfigurasi *database MySQL server slave*.

Pada pengaturan *server slave*, diisikan parameter pada *file* konfigurasi (CNF) *database MySQL* dalam *server slave*, seperti terlihat pada Gambar 11. Parameter replikasi yang digunakan adalah *REPLICATE\_DO\_DB* dengan nama “*uploadfile*” sebagai nama *database* yang akan digunakan menduplikasi alamat *database* pada *server master*. Setelah melakukan konfigurasi *database* pada *server slave*, *service* dalam *database* di-*restart* untuk memastikan parameter dapat berjalan dengan baik. Lalu, data di-*restore* dari *server master* ke *server slave*. Untuk keseluruhan proses yang disebutkan diatas, digunakan *tools PV*, yang juga dapat menampilkan proses dan waktu *restore database*.

```
server@server:~$ pv -W uploadfile-23.sql.gz | gunzip | mysql -uroot -p
1.14k18 0:00:00 [ 3720B/s ] [-----] 100%
server@server:~$ mysql -uroot -p
Enter password:
server@server:~$ mysql -uroot -p
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| uploadfile |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Gambar 12: Proses *restore database* dari *server master* kepada *server slave*

```
server@server:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| uploadfile |
+-----+
5 rows in set (0.07 sec)

mysql>
```

Gambar 13: Detail *database MySQL* pada *server slave*

```
mysql> change master to master_host='192.168.0.20', master_port=3306, master_user='rep1', master_password='replicate', master_auto_position=1;
Query OK, 0 rows affected, 2 warnings (0.01 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G
```

Gambar 14: Konfigurasi replikasi *database slave* ke *master*

Selanjutnya, Gambar 12 menunjukkan *proses restore database* dari *server master*. Proses pada Gambar 12 tergantung dari besaran data yang akan di-*restore*. Semakin besar *file size* dari data, maka akan semakin lama proses *restore database*. Lalu, dilakukan pengecekan dalam *database* yang di-*restore* (*server slave*). Gambar 13 menunjukkan proses *restore* yang berhasil dilakukan dari *database server master* ke *database server slave*. Setelah berhasil melakukan *restore*, langkah selanjutnya adalah melakukan konfigurasi untuk menyambungkan antara *database server slave* dengan *database server master* agar proses replikasi *database* dapat berjalan dengan semestinya.

Gambar 14 menunjukkan konfigurasi dari *database server slave* agar sistem replikasi dapat mengenali *database server slave*. Gambar 14 juga menunjukkan beberapa parameter user replika pada *server master*, yaitu *username*, *password*, *port database*, dan *host database*. Setelah selesai menjalankan perintah replikasi, maka tiba saatnya untuk menjalankan *server slave* dengan perintah “*start slave*”. Setelah *server slave* aktif, status replikasi *database server slave* pada log MySQL *server slave* dipastikan tidak memberikan *error*.

Log replikasi dari MySQL pada *server slave* ditunjukkan pada Gambar 15. Log tersebut menampilkan bahwa replikasi berjalan semestinya. Sampai pada langkah ini, konfigurasi pada *database MySQL server slave* telah selesai dan berlanjut pada p-ISSN:1693 – 2951; e-ISSN: 2503-2372



pengujian data yang akan dimasukkan ke dalam *database server master* dan *database server slave*.

```
mysql> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.0.20
Master_User: repl
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: masterreplicate-bin.000017
Read_Master_Log_Pos: 194
Relay_Log_File: slavereplicate-relay.000016
Relay_Log_Pos: 427
Relay_Master_Log_File: masterreplicate-bin.000017
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: uploadfile
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 194
Relay_Log_Space: 679
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
```

Gambar 15: Log replikasi MySQL *server slave*

C. Hasil Uji Coba Unggah Data

Data pada penelitian ini menggunakan beberapa *file*, yaitu *file* berformat PDF, *file* gambar IMG dan multimedia MP4. Besaran *file* yang digunakan pada penelitian ini adalah maksimal 100MB dengan banyak variasi kapasitas dari *file* tersebut. Terdapat 20 data yang berupa *file* PDF, *file* IMG dan *file* MKV.

Setelah data dimasukkan kedalam *database server master*, lalu dilakukan pengecekan di dalam *database server slave*. Jika data yang dimasukkan di dalam *database server master* ada pada *database server slave*, maka dipastikan replikasi berjalan sesuai.

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue 23 Aug 2022 12:27:25 PM UTC

System load:  0.07          Processes:            119
Usage of /:   13.3% of 48.96GB Users logged in:     0
Memory usage: 45%         IPv4 address for docker0: 172.17.0.1
Swap usage:   0%           IPv4 address for enp0s3: 192.168.0.20

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Aug 23 12:15:34 2022 from 192.168.0.105
server@servermaster:~$
```

Gambar 16: Status *server master*

```
server@servermaster:~$ sudo docker ps -a
[sudo] password for server:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
04081c6870e   mysql:5.7.22  "docker-entrypoint.s..." 4 months ago  Up 26 minutes  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp  mysql-master

server@servermaster:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| uploadfile |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Gambar 17: Status *database* MySQL *server master*

```
server@servermaster:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use uploadfile;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from tuploadfile;
+-----+-----+-----+-----+-----+-----+
| IDUploadFile | FilePath |
+-----+-----+-----+-----+-----+-----+
| 23 | C:\Users\Admin\Documents\dataset\Capture9.PNG |
| 24 | C:\Users\Admin\Documents\dataset\Capture10.PNG |
| 25 | C:\Users\Admin\Documents\dataset\Capture11.PNG |
| 26 | C:\Users\Admin\Documents\dataset\Capture12.PNG |
| 27 | C:\Users\Admin\Documents\dataset\Capture13.PNG |
| 28 | C:\Users\Admin\Documents\dataset\CLOSER_2017_39.pdf |
| 29 | C:\Users\Admin\Documents\dataset\Design and Implementation of a MySQL Database Backup.pdf |
| 30 | C:\Users\Admin\Documents\dataset\Nikolay_Baychenko.pdf |
| 31 | C:\Users\Admin\Documents\dataset\ONLampMySQLRepl.pdf |
| 32 | C:\Users\Admin\Documents\dataset\Knewella - Live For The Night.mkv |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Gambar 18: Isi *file* pada *database server master*

```
server@servermaster:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 32
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use uploadfile;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from uploadfile;
+-----+-----+
| IDUploadFile | FilePath |
+-----+-----+
| 23 | C:\Users\Admin\Documents\dataset\Capture9.PNG |
| 24 | C:\Users\Admin\Documents\dataset\Capture10.PNG |
| 25 | C:\Users\Admin\Documents\dataset\Capture11.PNG |
| 26 | C:\Users\Admin\Documents\dataset\Capture12.PNG |
| 27 | C:\Users\Admin\Documents\dataset\Capture13.PNG |
| 28 | C:\Users\Admin\Documents\dataset\CLOSER_2017_39.pdf |
| 29 | C:\Users\Admin\Documents\dataset\Design and Implementation of a MySQL Database Backup.pdf |
| 30 | C:\Users\Admin\Documents\dataset\Nikolay_Baychenko.pdf |
| 31 | C:\Users\Admin\Documents\dataset\ONLampMySQLRep1.pdf |
| 32 | C:\Users\Admin\Documents\dataset\Krewella - Live For The Night.mkv |
| 33 | C:\Users\Admin\Documents\dataset\Capture2.PNG |
| 34 | C:\Users\Admin\Desktop\29-Article Text-241-1-10-20210611.pdf |
| 35 | C:\Users\Admin\Desktop\123.txt.txt |
| 36 | C:\Users\Admin\Documents\dataset\Capture1.PNG |
| 37 | C:\Users\Admin\Documents\dataset\Capture3.PNG |
| 38 | C:\Users\Admin\Documents\dataset\Capture4.PNG |
| 39 | C:\Users\Admin\Documents\dataset\Capture5.PNG |
| 40 | C:\Users\Admin\Documents\dataset\Capture6.PNG |
| 41 | C:\Users\Admin\Documents\dataset\Capture7.PNG |
| 42 | C:\Users\Admin\Documents\dataset\Capture8.PNG |
+-----+-----+
20 rows in set (0.00 sec)

mysql>
```

Gambar 19: Data pada database server master

```
server@serverslave:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.7.22-log MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use uploadfile;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from uploadfile;
+-----+-----+
| IDUploadFile | FilePath |
+-----+-----+
| 23 | C:\Users\Admin\Documents\dataset\Capture9.PNG |
| 24 | C:\Users\Admin\Documents\dataset\Capture10.PNG |
| 25 | C:\Users\Admin\Documents\dataset\Capture11.PNG |
| 26 | C:\Users\Admin\Documents\dataset\Capture12.PNG |
| 27 | C:\Users\Admin\Documents\dataset\Capture13.PNG |
| 28 | C:\Users\Admin\Documents\dataset\CLOSER_2017_39.pdf |
| 29 | C:\Users\Admin\Documents\dataset\Design and Implementation of a MySQL Database Backup.pdf |
| 30 | C:\Users\Admin\Documents\dataset\Nikolay_Baychenko.pdf |
| 31 | C:\Users\Admin\Documents\dataset\ONLampMySQLRep1.pdf |
| 32 | C:\Users\Admin\Documents\dataset\Krewella - Live For The Night.mkv |
| 33 | C:\Users\Admin\Documents\dataset\Capture2.PNG |
| 34 | C:\Users\Admin\Desktop\29-Article Text-241-1-10-20210611.pdf |
| 35 | C:\Users\Admin\Desktop\123.txt.txt |
| 36 | C:\Users\Admin\Documents\dataset\Capture1.PNG |
| 37 | C:\Users\Admin\Documents\dataset\Capture3.PNG |
| 38 | C:\Users\Admin\Documents\dataset\Capture4.PNG |
| 39 | C:\Users\Admin\Documents\dataset\Capture5.PNG |
| 40 | C:\Users\Admin\Documents\dataset\Capture6.PNG |
| 41 | C:\Users\Admin\Documents\dataset\Capture7.PNG |
| 42 | C:\Users\Admin\Documents\dataset\Capture8.PNG |
+-----+-----+
20 rows in set (0.00 sec)

mysql>
```

Gambar 20: Isi dari database server slave yang sudah sesuai dengan isi dari database server master

Setelah menyiapkan data, disiapkan server master. Sebelum mengisi data kedalam database server master, dipastikan kondisi server master terlebih dahulu. Gambar 16 menunjukkan status server master. Setelah melakukan pengecekan kondisi server master, dilakukan pengecekan pada status database MySQL. Pengecekan sebelum melakukan

pengujian berfungsi untuk memastikan agar tidak terjadi kegagalan saat proses unggah data.

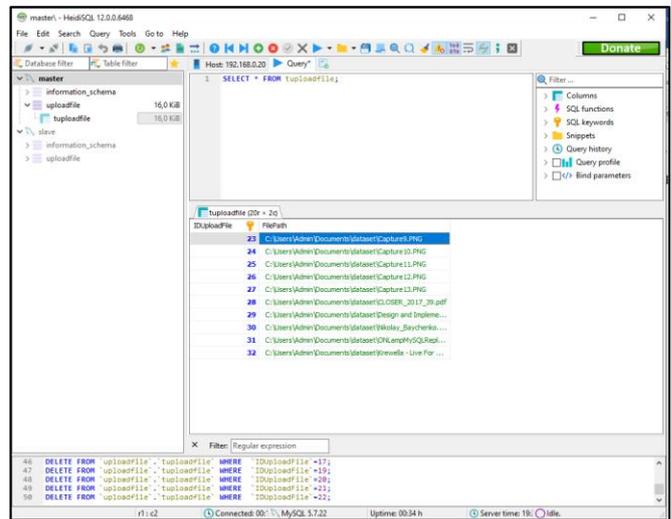
Gambar 17 menunjukkan status dari MySQL pada database server master. Gambar 17 menunjukkan bahwa database MySQL sudah siap untuk dilakukan unggahan data. Gambar 18 menunjukkan percobaan melakukan unggah data. Pada gambar tersebut, terlihat terdapat 11 data dalam percobaan unggah data. Dari hasil query database, ditunjukkan bahwa unggah data ke dalam server master telah berhasil dilakukan.

Lalu pada Gambar 19, ditunjukkan bahwa data telah terunggah pada database server master. Terhitung total 20 file telah terunggah ke dalam database server master. Total 20 file yang diunggah pada database server master ini juga telah terreplikasi pada database server slave.

Gambar 20 menunjukkan bukti bahwa 20 data yang diunggah pada database server master telah terreplikasi pada database server slave. Perlu dilakukan pengujian lebih lanjut, yaitu dengan pengujian hapus data. Apabila salah satu atau lebih data dihapus pada database server master, maka seharusnya data tersebut juga terhapuskan pada database server slave.

#### D. Hasil Uji Coba Hapus Data

Setelah melakukan pengujian unggah data, selanjutnya dilakukan pengujian menghapus data pada data pengujian. Diharapkan dari pengujian hapus dataset adalah mengetahui seberapa berfungsi replikasi data pada database MySQL. Untuk melakukan penghapusan data, digunakan tools bantuan SQL editor, yaitu HeidiSQL. Menggunakan SQL editor untuk mengelola data pada database dirasa dapat membantu oleh karena ketersediaan Graphical User Interface (GUI) daripada CLI. Uji coba penghapusan data dimulai dengan menghapus 10 data pada database server master.



Gambar 21: Proses hapus data pada database server master



Gambar 21 menunjukkan penghapusan 10 data dan menyisakan 10 data pada *database server master*. Setelah dicek, *database* pada *server slave* juga menyisakan 10 data yang sama, seolah penghapusan 10 data pada *database server master* sebelumnya juga terjadi pada *database* pada *server slave*.

### E. Analisis Hasil dan Pembahasan

Berdasarkan hasil pengujian penelitian replikasi database menggunakan MySQL menunjukkan bahwa replikasi database *master slave* sesuai dengan teori dari replikasi *database master slave* MySQL. Hasil ini serupa dengan penelitian milik Noor Kamilah Malik, dkk yang menjelaskan mengenai fungsi dan juga cara kerja dari sistem tersebut [19]. Hasil pengujian menunjukkan bahwa replikasi mendapatkan hasil yang sesuai dengan metode uji, seperti unggah data ke dalam *database* dan menghapus data. Total terdapat 20 data yang dapat direplikasi, baik dalam proses unggah data ataupun proses penghapusan data.

Pembahasan berikutnya tentang replikasi *master slave* juga memiliki kelemahan, yaitu jika terlalu lama *server slave* tidak bisa terhubung dengan *server master*, maka akan terjadi *error*. *Error* yang terjadi dikarenakan *server* tidak dapat melakukan pengecekan keaktualan data, maka secara otomatis *database* yang ada pada *server slave* akan melakukan pemutusan dengan kode *error* yang muncul ketika dilihat pada *log* status *slave*. Berikut adalah hasil *error* dari *server slave* yang diakibatkan *server slave* tidak dapat melakukan pengecekan data ke *server master*.

```
mysql> show slave status\G;
+-----+
| Slave_IO_State:                               |
| Master_Host: master1                          |
| Master_User: rep                              |
| Master_Port: 3306                             |
| Connect_Retry: 60                             |
| Master_Log_File:                              |
| Read_Master_Log_Pos: 4                        |
| Relay_Log_File: mysql-slave2-relay-bin-rep002dmaster1.000002 |
| Relay_Log_Pos: 4                              |
| Relay_Master_Log_File:                       |
| Slave_IO_Running: No                          |
| Slave_SQL_Running: Yes                       |
| Replicate_Do_DB:                             |
| Repligate_Ignore_DB:                         |
| Until_Log_File:                              |
| Until_Log_Pos: 0                             |
| Master_SSL_Allowed: No                       |
| Master_SSL_CA_File:                          |
| Master_SSL_CA_Path:                          |
| Master_SSL_Cert:                             |
| Master_SSL_Cipher:                           |
| Master_SSL_Key:                              |
| Seconds_Behind_Master: 0                     |
| Master_SSL_Verify_Server_Cert: No            |
| Last_IO_Errno: 1236                          |
| Last_IO_Error: Got fatal error 1236 from master when reading data from binary log: 'The slave is coming using CHANGE MASTER TO MASTER_AUTO_POSITION = 1, but the master has purged binary logs containing GTIDs that the slave requires.'
```

Gambar 22: Error pada *database* dalam *server slave*

Pada Gambar 22, ditunjukkan contoh *error* yang terjadi ketika *database server slave* gagal terhubung dengan *database server master* yang mengakibatkan posisi "binlog" dari *database slave* terputus. Penyebab kejadian putus koneksi antara *server slave* dengan *server master* yang paling umum terjadi adalah koneksi internet.

Jika terjadi *error* pada Gambar 23, maka yang harus dilakukan adalah melakukan *restore* ulang data dari *database master* ke dalam *database server slave* lalu dilakukan konfigurasi ulang dengan melakukan *set master*. Selain *error* yang disebabkan koneksi, *error* pada *server slave* juga dapat terjadi jika melakukan *input* data melalui *server slave*. Secara

teori, data yang masuk harus melalui *server master*, lalu dilakukan replikasi secara otomatis oleh MySQL. Jika terdapat data yang diunggah pada *server slave*, maka akan terjadi *error* seperti pada Gambar 24.

```
Exec_Master_Log_Pos: 632715
Relay_Log_Space: 263542678
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 1594
Last_SQL_Error: Relay log read failure: Could not parse relay log event entry. The possible reasons are: the master's binary log is corrupted (you can check this by running 'mysqbinlog' on the binary log), the slave's relay log is corrupted (you can check this by running 'mysqbinlog' on the relay log), a network problem, or a bug in the master's or slave's MySQL code. If you want to check the master's binary log or slave's relay log, you will be able to know their names by issuing 'SHOW SLAVE STATUS' on this slave.
```

Gambar 23: Error pada *database server slave*

Pada Gambar 23, ditunjukkan *error* pada *server slave* yang disebabkan oleh data yang masuk melalui *server slave*. *Error* tersebut terjadi karena MySQL *server slave* melakukan pengecekan data dan terjadi kerusakan pada *binary log* sehingga proses replikasi dihentikan secara otomatis oleh *server slave*. Solusi dari *error* tersebut adalah dilakukan *restore* ulang dari *database server master* dan diulang melakukan *set master*.

Namun, dengan adanya kelemahan tersebut, keunggulan yang dimiliki tetap memiliki pengaruh yang signifikan bagi perusahaan. Sistem yang langsung bisa digunakan oleh *end user* bisa mempermudah pekerja dalam melakukan pekerjaannya. Hal tersebut sejalan dengan studi dari Efri Darwis yang menyatakan bahwa penggunaan sistem ini berjalan dengan cepat sehingga efektif jika bisa dimanfaatkan dengan baik [20].

## V. KESIMPULAN

Simpulan dari pengujian replikasi *database* menggunakan metode *master-slave* sangat layak digunakan pada perusahaan. Pasalnya, jika terjadi masalah, maka solusinya yakni dengan *restore* ulang dari *database server master* dan diulang menggunakan *set master*. Pengimplementasian tersebut tidak perlu menggunakan orang yang ahli dalam bidang IT. Sehingga, ide ini bisa digunakan oleh perusahaan yang belum memiliki staff atau karyawan yang ahli di bidang IT. Penelitian ini bisa digunakan untuk mempermudah sistem di perusahaan karena efektivitas penggunaan dan hanya dengan sekali setup. Proses konfigurasi replikasi *master-slave* lebih mudah jika dibandingkan dengan konsep *master-master* atau *cluster*. Konsep yang ditawarkan dari replikasi ini juga sangat sederhana, sehingga memudahkan *end user*.

## REFERENSI

- [1] M. Johnson *et al.*, "Impact of big data and artificial intelligence on industry: Developing a workforce roadmap for a data driven economy," *Glob. J. Flex. Syst. Manag.*, vol. 22, no. 3, hal. 197–217, 2021.
- [2] G. Aceto, V. Persico, dan A. Pescapé, "Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0," *J. Ind. Inf. Integr.*, vol. 18, hal. 100129, 2020.
- [3] F. Zamzami, N. D. Nusa, dan I. A. Faiz, *Sistem Informasi Akuntansi*. UGM PRESS, 2021.
- [4] S. Arni, "Penerapan Perintah MySQL dalam Backup Database,"

Imiyatus Sholihah: Sistem Replikasi Basis Data...

- PROGRES, vol. 5, no. 2, hal. 19–26, 2013.
- [5] A. Meier dan M. Kaufmann, *SQL & NoSQL databases*. Springer, 2019.
- [6] H. Maulana, “Analisis Dan Perancangan Sistem Replikasi Database MySQL dengan menggunakan VMware pada Sistem Operasi Open Source,” *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 1, no. 1, hal. 32–37, 2016.
- [7] Z. Jiang, H. Da, Y. Qiu, dan J. Pan, “Overview of Data Synchronization and Fault Recovery Technology in Multi Active Data Center,” in *2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, 2021, hal. 113–118.
- [8] R. Shrestha, “High Availability and Performance of Database in the Cloud-Traditional Master-slave Replication versus Modern Cluster-based Solutions,” 2017.
- [9] Z. M. Subekti, S. Subandri, dan G. Rakasiwi, “Perancangan Infrastruktur Web Server dan Database menggunakan Metode Replication Mirror dan Failover Clustering,” *J. Cendikia*, vol. 18, no. 1, hal. 359–370, 2019.
- [10] A. Gupta, M. Saxena, dan R. Gill, “Performance Analysis of RDBMS and Hadoop Components with their File Formats for the development of Recommender Systems,” in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, hal. 1–6.
- [11] I. M. Sukarsa, “Aplikasi Konversi Flowchart ke Kode Program Bahasa Pemrograman PL/SQL MYSQL,” *Maj. Ilm. Teknol. Elektro*, vol. 8, no. 2, hal. 44–53, 2009.
- [12] F. Y. Putra, I. M. A. Suyadnya, dan I. P. R. Agung, “Reservation Application System Of Private Lesson At Easyspeak Denpasar Based On Web And Android,” *Maj. Ilm. Teknol. Elektro*, vol. 15, no. 1, hal. 79–83, Jun 2016, doi: 10.24843/MITE.1501.14.
- [13] X. Yu, X. Cai, S. Su, J. Han, dan R. F. Mansour, “The Technical Design and Implementation of Cross-Platform Industrial Product Order System,” *J. Cybersecurity*, vol. 3, no. 1, hal. 1, 2021.
- [14] A. Amarudin dan Y. Atri, “Analisis Penerapan Mikrotik Router Sebagai User Manager Untuk Menciptakan Internet Sehat Menggunakan Simulasi Virtual Machine,” *J. TAM (Technology Accept. Model.)*, vol. 9, no. 1, hal. 62–66, 2018.
- [15] Z. Aalam, V. Kumar, dan S. Gour, “A review paper on hypervisor and virtual machine security,” in *Journal of Physics: Conference Series*, 2021, vol. 1950, no. 1, hal. 12027.
- [16] M. Idhom, R. Alit, H. Endah, dan A. Fauzi, “Implementation System Telegram Bot for Monitoring Linux Server,” 2018.
- [17] L. Mercl dan J. Pavlik, “The Comparison of Container Orchestrators,” in *Third International Congress on Information and Communication Technology*, 2019, hal. 677–685.
- [18] Y. C. Giap *et al.*, *Cloud Computing: Teori dan Implementasi*. Yayasan Kita Menulis, 2020.
- [19] N. K. Malik, A. T. P. A. Ramadani, N. A. Karlina, dan D. R. Lestyarini, “Replikasi Database dengan Model Master-Master dan Master-Slave MariaDB,” Universitas Siliwangi, 2020.
- [20] E. Darwis, “Implementasi Basis Data Terdistribusi menggunakan MySQL pada PT Thamrin Brothers Palembang,” Universitas Bina Darma, 2012.



{ halaman ini sengaja di kosongkan }