

# UJI COBA MODEL REPLIKASI ASYNCHRONOUS DATABASE TERPUSAT DI LEMBAGA PELATIHAN KOMPUTER

**Ari Muzakir**

Ilmu Komputer Universitas Bina Darma Palembang  
Jl A. Yani No.03, Plaju, Palembang, Sumatera Selatan 30257  
Email : ariemuzakir@gmail.com

**Abstrak** – Tujuan dari penelitian ini adalah untuk menguji coba model replikasi secara asynchronous pada database terpusat pada lembaga pelatihan komputer Sungai Lilin. Untuk merancang arsitektur database terpusat ini memanfaatkan data administrasi yang ada di lembaga pelatihan komputer. Dimana, pada lembaga pelatihan komputer di daerah Sungai Lilin ini memiliki 3 cabang pembantu yang terus menerus membutuhkan update data dari sistem administrasinya. Kemudian seringnya pemadaman dari PLN menyebabkan proses update data menjadi terganggu. Pada replikasi asynchronous, mesin slave tidak perlu dihubungkan secara permanen untuk menerima update dari database master. Dengan menggunakan model replikasi asynchronous database terpusat ini, maka kemungkinan akan adanya kehilangan data atau kerusakan data akan dapat di minimalisasi. Proses replikasi asynchronous dilakukan secara real time dan terus menerus sehingga jika terjadi error atau kehilangan sumber daya (listrik) pada salah satu slave maka data akan langsung dialihkan ke komputer slave yang aktif. Pada pengujian database ini menggunakan engine mysql dan memanfaatkan konfigurasi pada phpmyadmin yang terinstall melalui aplikasi XAMPP. Hasil dari konfigurasi harus bernilai true pada master dan slave sehingga proses replikasi dapat dilakukan. Proses replikasi memanfaatkan alamat internet protocol (IP) dari masing-masing komputer.

**Kata kunci:** Model Replikasi, Replikasi Asynchronous, Database Terpusat

## I. PENDAHULUAN

Pada era perkembangan teknologi yang semakin pesat serta ketatnya persaingan bisnis global yang ada saat ini, maka banyak perusahaan dan organisasi yang menerapkan teknologi informasi untuk mendukung perkembangan perusahaan, organisasi, maupun institusi. Dengan memanfaatkan teknologi informasi tersebut diharapkan dapat mempermudah proses bisnis perusahaan serta meningkatkan efisiensi dari kegiatan rutin. Pengolahan data, informasi dan sumber daya perusahaan dengan menggunakan teknologi informasi dapat membantu untuk meminimalkan pengeluaran perusahaan untuk biaya operasional. Bagi perusahaan yang memiliki banyak cabang kebutuhan adanya basis data yang mampu menampung data dengan jumlah yang banyak sangat diperlukan.

Basis data sentralisasi adalah sistem basis data yang dikontrol secara tersentralisasi atau bisa juga dikatakan hanya ada satu pengontrol, [2]. Sistem tersebut berjalan pada sistem komputer tunggal dan berinteraksi dengan komputer yang lain. Sentralisasi sendiri merupakan sistem yang menempatkan data informasi yang terpusat yang umumnya bersifat tertutup untuk umum. Dalam distribusi terpusat mainframe berfungsi sebagai sumber basis data dan intranet yang sangat penting. Untuk ciri dari pendekatan terpusat adalah termasuk efisiensi kontrol, dan ekonomi. Dapat diperumpamakan sebuah sistem pengarsipan terpusat, dimana catatan untuk beberapa

orang dan unit yang terletak di salah satu lokasi pusat dan di kendali oleh seorang staf dalam pencatatannya. Dalam kasus besar sistem pengarsipan terpusat itu bisa dikatatan pengiriman data yang terpusat. Hal ini yang menjadi permasalahan disistem pendataan administrasi Sungai Lilin.

Permasalahan yang ada disistem pendataan administrasi saat ini belum mempunyai arsitektur standar yang mengatur pola informasi data dari cabang kepusat ataupun sebaliknya. Sinkronisasi yang dilakukan masih menggunakan cara manual, yaitu transfer data masih menggunakan media *flash disk* atau *compact disk*. Jika data diperlukan maka harus diantar dari cabang secara langsung ke pusat atau dikirim melalui via internet menggunakan e-mail dan begitu juga sebaliknya. sehingga timbul permasalahan tidak cepatnya informasi yang di dapat oleh pusat dan kesinkronan data tidak terjamin, dengan kata lain tiap cabang belum menerapkan konsep basis data sentralisasi yang terpusat. Adanya model replikasi data juga belum dimanfaatkan untuk dapat menjamin ketersediaan data di masing-masing cabang, yang didukung dengan sebuah program atau aplikasi dengan kualitas yang memadai dalam basis datanya. Sehingga sulit dalam akses data yang meliputi pengolahan, penyimpanan dan pengeditan data.

Jika sesuatu relasi  $r$  direplikasi maka sebuah salinan dari relasi disimpan pada dua atau lebih lokasi. Dalam kasus yang ekstrem, kita bias juga memiliki replikasi penuh, dimana sebuah salinan dari relasi  $r$

dapat disimpan pada setiap lokasi dalam sistem. Dengan demikian menggunakan teknik replikasi ini data dapat didistribusikan ke lokasi yang berbeda tempat maupun fisik, melalui koneksi jaringan lokal maupun jaringan internet. Sistem replikasi memungkinkan untuk mendukung kinerja pada program yang telah dibuat, untuk penyebaran data fisik sesuai dengan bagaimana mengaplikasikannya. Dalam model replikasi ini terdiri dari *one master, one slave, one master, many slave* [3].

Cara kerja teknologi replikasi data memiliki fungsi yang rumit karena secara otomatis menyalin data ke lokasi yang ditentukan, setelah data yang lengkap sudah direplikasi ke target yang dimaksud, maka hanya data yang berubah yang akan direplikasi, dan sehingga akan menghemat kebutuhan bandwidth dalam prosesnya. Data hasil dari replikasi yang ada di penyimpanan biasa disebut sebagai *seeding*. Setelah data yang di *master* telah di *seeding*, maka fungsi replikasi berikutnya dapat berjalan pada dua mode *synchronous* dan *asynchronous*.

Replikasi *asynchronous* merupakan salah satu jenis replikasi basis data. Pada replikasi *asynchronous* mesin *slave* replikasi tidak perlu dihubungkan secara permanen untuk menerima *update* dari basis data *master*. Hal ini berarti bahwa pembaruan dapat terjadi melalui sambungan jarak jauh dan bahkan solusi sementara seperti layanan dial-up [4]. Prinsip kerja metode replikasi *asynchronous* yaitu setiap basis data *master* akan membuat *log file* dalam proses inputnya, sedangkan *log file* sendiri berisi seluruh aktivitas yang terjadi pada basis data *master* baik berupa input, *update* ataupun *delete* terhadap basis datanya. Pada suatu saat ketika jaringan basis data *slave* terhubung dengan database *master*, maka secara otomatis basis data *slave* akan melihat aktivitas yang ada pada *log file* dan akan meng-*update* atau menyamakan dengan perubahan yang terjadi pada basis data *master*.

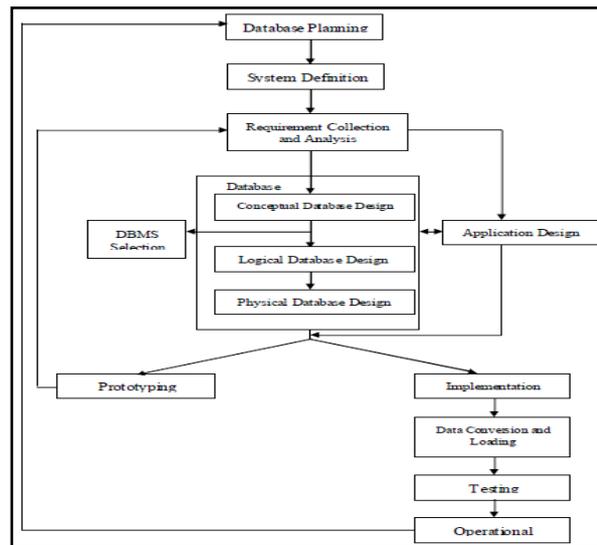
Solusi dari permasalahan diatas yaitu membangun sistem basis data sentralisasi yang menerapkan replikasi *asynchronous*. Sistem ini client mengirim atau menginput data ke basis data pusat (*master*), lalu secara otomatis cabang atau (*slave*) meng-*update* data dari pusat. Untuk media penyimpanan pada basis data pusat, basis data akan dibagi menjadi dua yaitu *master* dan *slave* menggunakan metode replikasi *asynchronous*. Metode replikasi yang terdapat pada MySQL adalah *asynchronous*, yaitu *server* dengan type *slave* tidak selalu harus terkoneksi secara permanen untuk menerima berbagai *update* pada basis data *server master*.

## 2. PEMBAHASAN

Metode yang digunakan dalam penelitian ini adalah metode deskriptif. Dimana metode deskriptif yaitu metode mengemukakan masalah dengan mengumpulkan data dan menyajikan data terhadap

suatu objek penelitian, yang bertujuan untuk mengambil suatu kesimpulan dari pembahasan yang dilakukan.

Adapun Untuk merancang desain sistem basis data menggunakan metode *Database Life Cycle* (DBLC). Diperlukan tahapan-tahapan terstruktur yang harus diikuti, Tahapan tersebut dinamakan dengan Siklus Hidup Basis Data (*Database Lifecycle*) [1]. Tahapan yang terdapat dalam DBLC tidak harus berurutan, namun juga melibatkan beberapa pengulangan ke tahapan sebelumnya melalui putaran balik. Tahapan tersebut terlihat seperti gambar 1.



Gambar 1. Tahapan DBLC

### 2.1 Database Planning

Perancangan basis data bertujuan untuk menyimpan data dan mengolah data. Perancangan basis data dimaksud untuk memberikan kemudahan pelayanan data terhadap aktifitas sistem administrasi dimana data arsip yang akan disimpan di dalam sebuah basis data akan memberikan kemudahan dalam proses pengolahan data dan pencarian data arsip untuk menunjang kelancaran proses sistem lembaga pelatihan komputer sendiri. Adapun kebutuhan akan proses basis data antara lain untuk mengelola (memasukkan, memperbarui, dan menghapus) data-data. Selain itu juga dapat dilakukan pencarian data yang berhubungan dengan sistem yang ada di lembaga pelatihan.

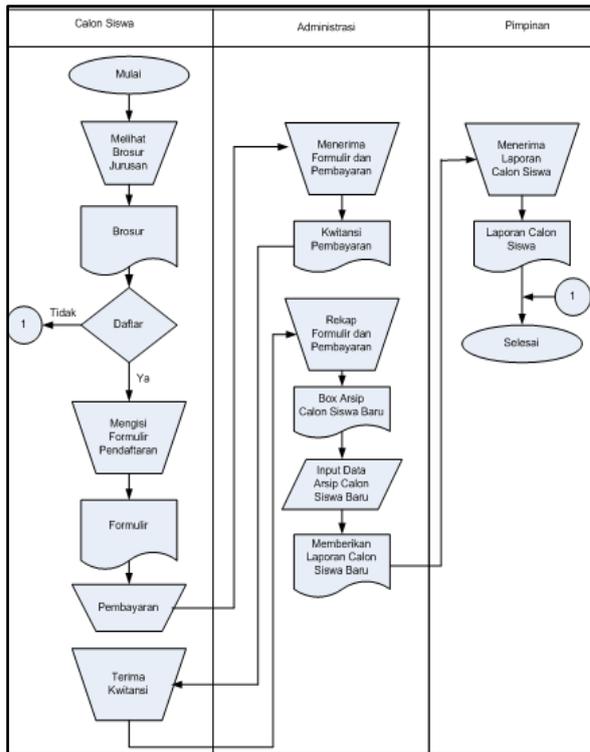
### 2.2 System Definition

Definisi sistem merupakan alur dari sistem yang mencakup data-data arsip yang ada pada sistem lembaga pelatihan komputer di Sungai Lilin, dimana definisi sistem yang ada saling berkaitan dan skema yang ada mencakup pendaftaran penerimaan siswa baru, data pemilihan minat dan data pembayaran, berkas yang didaftarkan berhubungan dengan arsip, arsip tersebut kemudian berhubungan dengan kotak penyimpanan dan arsip tersebut berhubungan dengan laporan, laporan dilihat oleh pimpinan. Pimpinan menerima laporan dari administrasi dan melihat

laporan siswa baru dan pembayaran yang masuk untuk periode tertentu.

**2.3 Requirement collection and analysis**

Pada tahapan ini dilakukan pengumpulan data serta analisa informasi mengenai sistem yang berjalan di lembaga pelatihan komputer Sungai Lilin. Berikut diperlihatkan pada gambar 2 sistem yang berjalan saat ini.



**Gambar 2. Flowchart Sistem Berjalan saat ini**

Informasi tersebut selanjutnya dilakukan analisa untuk mengidentifikasi kebutuhan pengguna dan diharapkan nantinya dapat digunakan pada sistem yang menggunakan basis data baru. Hasil dari analisa tersebut dapat disajikan dalam bentuk tabel 1 berikut.

**Tabel 1. Analisa dan pengumpulan data kebutuhan**

Data	Tipe Akses	Administrasi	Kepala Kantor
Siswa Baru	Query	X	
	View	X	
	Surat		
	Laporan		X
Jurusan	Query	X	
	View	X	
	Surat		
	Laporan		X
Pembayaran	Query	X	
	View	X	
	Surat		
	Laporan		X
Administrasi	Query	X	

	View	X	
	Surat		
	Laporan		
Replikasi	Query	X	
	View	X	
	Surat		

**2.4 Database Design**

Setelah tahapan pengumpulan kebutuhan dan analisis sistem yang saat ini digunakan atau berjalan, maka didapatkan beberapa bentuk desain tabel. Berikut desain dari basis data ditunjukkan pada tabel 2 yang berisi atribut dari tipe entitas dalam sistem ini.

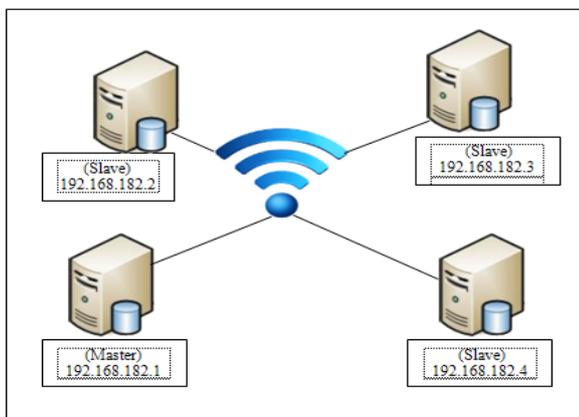
**Tabel 2. Tabel Atribut dari Tipe Entitas**

No	Nama Entitas	Atribut	Deskripsi	Tipe Data & Ukuran
1	Siswa Baru	No_daftar	No Pendaftaran	Int(8)
		Nm_siswa_baru	Nama Siswa Baru	Varchar(25)
		Tmpt_lahir	Tempat Lahir	Varchar(20)
		Tgl_lahir	Tanggal Lahir	Date(8)
		Pendidikan	Pendidikan	Varchar(20)
		Alamat	Alamat	Varchar(20)
		Telepon	Telepon	Varchar(12)
		Id_jurusan	Id Jurusan	Int(4)
		Username	Username	Varchar(25)
Password	Passowrd	Varchar(25)		
2	Jurusan	Id_jurusan	Id Jurusan	Int(4)
		Nm_jurusan	Nama Jurusan	Varchar(25)
		Ket	Keterangan	Varchar(255)
3	Pembayaran	Id_Bayar	ID Pembayaran	Int(15)
		Tgl_bayar	Tanggal Bayar	Varchar(20)
		No_daftar	Nomor Daftar	Varchar(15)
		Biaya_daftar	Biaya Daftar	Int(15)
		Biaya_almamater	Biaya Almamater	Varchar(20)
		Biaya_bukudiklat	Biaya Buku Diklat	Varchar(15)

		Biaya_ordik	Biaya Ordik	Int(15)
		SPP	SPP	Varchar(20)
		Total Bayar	Total Bayar	Varchar(15)
8	Administrasi	Id_admin	Id administrasi	Int(4)
		Nama	Nama	Varchar(35)
		Username	Username	Varchar(25)
		Password	Password	Varchar(25)

### 3. Implementasi

Hasil dari penelitian ini akan diujicobakan menggunakan suatu aplikasi sederhana berbasis *web* dimana dapat dijalankan menggunakan jaringan *intranet* maupun *internet*. Titik berat dari penelitian ini yaitu pada model replikasi data dan proses *asynchronisasi* data antar cabang ke pusat. Berikut arsitektur dari sistem sederhana yang dibangun yang ditunjukkan pada gambar 3.



Gambar 3. Arsitektur Sistem Replikasi Data

Penjelasan dari Gambar tersebut yaitu sebagai berikut.

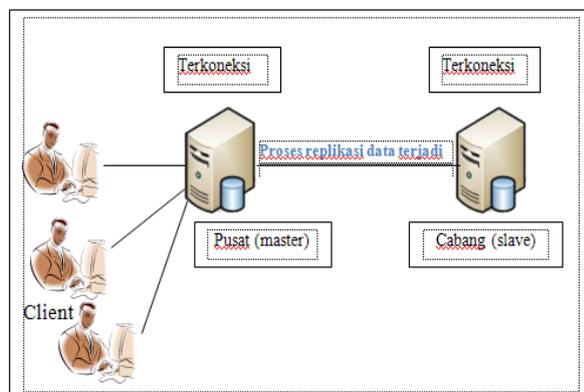
1. Proses entry data akan dilakukan oleh admin atau user pada setiap cabang, data yang sudah di *update* akan disimpan ke basis data pusat kemudian akan tereplikasi secara otomatis database *slave*.
2. Setiap basis data cabang melakukan *update* data, maka untuk data yang sama yang ada di basis data pusat otomatis akan ikut ter-*update* juga. Dengan begitu data pada basis data cabang akan sama dengan basis data pusat
3. Jika pada saat tertentu basis data cabang bermasalah atau koneksi terputus, maka data yang akan didistribusikan akan tertampung terlebih dahulu di basis data lokal. Ketika *server* basis data

pusat sudah kembali normal, data bisa didistribusikan kembali

4. Pada basis data pusat yang menjadi *server* utama akan digunakan dua *server* yaitu *master* dan *slave*. *Slave* berfungsi untuk menyimpan *backup* data yang berada di *master*.

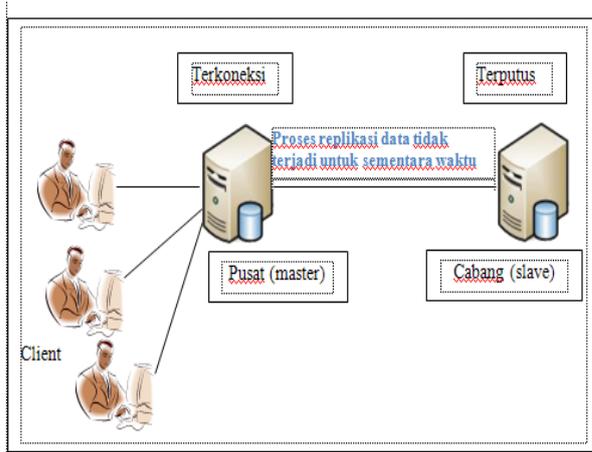
Pada saat proses koneksi yang melibatkan cabang dan pusat terdapat 3 kondisi yang akan terjadi :

1. *Server* Cabang dan Pusat Terkoneksi. Dalam kondisi ini, proses replikasi dapat berjalan dengan baik dikarenakan *server* pusat dan cabang tidak bermasalah. Sehingga kondisi ini dapat dikatakan kondisi paling aman dalam proses pendistribusian data dikarenakan kedua *server* berjalan sebagaimana mestinya. Untuk gambaran umum diperlihatkan pada gambar 4 berikut.



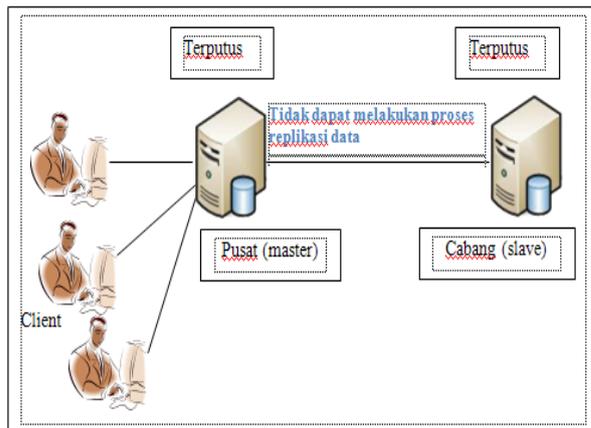
Gambar 4. Server cabang dan pusat terkoneksi

2. *Server Master* hidup tetapi *Slave* Mati. Dalam kondisi kedua yang dapat terjadi jika *server Master* hidup, tetapi *slave* mati atau ada kendala dengan internet. Untuk menangani masalah ini *master* harus melakukan proses pengolahan data pada basis data lokalnya dahulu. Jadi meskipun tidak terhubung dengan *slave*, pusat masih bisa melakukan proses pengolahan data dengan status pengiriman data yang belum di *update* ke *slave*. Sehingga didalam tabel sementara terdapat status pengiriman data ke *slave* apakah sudah berhasil di *update* atau belum, dan ketika *slave* terhubung maka akan secara otomatis data terplikasi ke *slave* seperti yang diperlihatkan pada gambar 5.



Gambar 5. Server master hidup tetapi slave mati

3. *Server (master) Mati dan Slave Mati*. Dengan kondisi seperti ini, proses replikasi tidak dapat dilakukan. Karena *server* cabang dan pusat sedang bermasalah dan proses koneksi antar cabang tidak dapat terjadi. Proses distribusi dapat terjadi jika *server* keduanya sudah berjalan normal lagi, dapat *input* data di *master* saja. Pada gambar 6 mensimulasikan keadaan pada saat *server master* dalam kondisi mati dan *slave* mati.



Gambar 6. Server master mati dan slave mati

Dalam penelitian ini model replikasi yang digunakan adalah replikasi data tersentralisasi. Replikasi adalah sebuah teknik untuk melakukan *copy* dan pendistribusian data dan objek-objek basis data dari satu basis data ke basis data lain dan melaksanakan sinkronisasi antara basis data sehingga konsistensi data dapat terjamin. Untuk replikasi *asynchronous* pada basis data mysql yaitu dimana salah satu ada yang bertindak sebagai komputer *master* dan yang lain sebagai komputer *slave*, Untuk cara kerja replikasi nantinya apabila data yang di komputer *master* dirubah maka otomatis di komputer *slave* juga akan berubah, dan begitu juga sebaliknya. Diasumsikan untuk IP komputer *master* adalah (192.168.182.1) sedangkan untuk komputer *slave* adalah (192.168.182.2).

Untuk itu diperlukan konfigurasi pada komputer *server (master)* seperti pada gambar 7 berikut ini.

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'slave'@'192.168.182.2' IDENTIFIED BY 'slave';
Query OK, 0 rows affected (0.19 sec)

mysql> RESET MASTER;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 1
Current database: *** NONE ***
Query OK, 0 rows affected (0.21 sec)

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 | 98      | db_replikasi |                    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> RESET MASTER;
Query OK, 0 rows affected (0.14 sec)

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 | 98      | db_replikasi |                    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 7. Konfigurasi pada komputer server

Selanjutnya adalah masuk ke directory C:/xampp/mysql/bin/my.cnf dan edit kode simpan file konfigurasi serta restart service MySQL. Konfigurasi ini dapat dilihat pada gambar 8.

```
# The following options will be passed to all MySQL clients
[client]
# password      = your_password
# port          = 3306
# socket        = "c:/xampp/mysql/mysql.sock"

# Here follows entries for some specific programs

# The MySQL server
[mysqld]
server-id = 1
log-bin = c:/xampp/mysql/mysql-bin.log
binlog-do-db = db_replikasi

port = 3306
socket = "c:/xampp/mysql/mysql.sock"
basedir = "c:/xampp/mysql"
tmpdir = "c:/xampp/tmp"
datadir = "c:/xampp/mysql/data"
skip-locking
key_buffer = 16M
max_allowed_packet = 1M
```

Gambar 8. Konfigurasi file my.in pada MySQL service

Kemudian perlu dilakukan konfigurasi pada komputer *Server 2 (Slave)*. Selanjutnya merubah kode simpan file konfigurasi serta me-restart service MySQL seperti pada gambar 9.

```
# The MySQL server
[mysqld]

server-id = 2
master-host = 192.168.182.1
master-user = slave
master-password = slave
master-connect-retry = 60
replicate-do-db = db_replikasi

port = 3306
socket = "c:/xampp/mysql/mysql.sock"
basedir = "c:/xampp/mysql"
tmpdir = "c:/xampp/tmp"
datadir = "c:/xampp/mysql/data"
skip-locking
key_buffer = 16M
max_allowed_packet = 1M
table_cache = 64
sort_buffer_size = 512K
net_buffer_length = 8K
read_buffer_size = 256K
read_rnd_buffer_size = 512K
mysam_sort_buffer_size = 8M
log_error = "mysql_error.log"
```

```
# commented in by lampp security
#skip-networking
skip-federated

# Replication Master Server (default)
# binary logging is required for replication
# log-bin deactivated by default since XAMPP 1.4.11
#log-bin=mysql-bin

# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id = 2

# Replication Slave (comment out master section to use this)
```

Gambar 9. Konfigurasi di sisi slave

Langkah terakhir adalah aktivasi replikasi dengan cara masuk ke terminal *slave* agar dapat dikoneksikan dengan *server*. Proses ini akan mengaktifasi fungsi replikasi di komputer *slave* yang ditunjukkan pada gambar 10 berikut.

```
mysql> SLAVE STOP;
Query OK, 0 rows affected (23.29 sec)

mysql> CHANGE MASTER TO MASTER_HOST='192.168.182.1',MASTER_USER='slave',MASTER_P
ASSWORD='slave',MASTER_LOG_FILE='mysql-bin.000013',MASTER_LOG_POS=2082;
Query OK, 0 rows affected (0.09 sec)

mysql> START SLAVE;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW SLAVE STATUS
+-----+-----+-----+-----+-----+-----+
| Slave_IO_State: | Waiting for master to send event |
| Master_Host:    | 192.168.182.1 |
| Master_User:    | slave |
| Master_Port:    | 3306 |
| Connect_Retry:  | 60 |
| Master_Log_File: | mysql-bin.000013 |
| Read_Master_Log_Pos: | 2082 |
| Relay_Log_File: | HP-NB-relay-bin.000002 |
| Relay_Log_Pos:  | 235 |
| Relay_Master_Log_File: | mysql-bin.000013 |
| Slave_IO_Running: | Yes |
| Slave_SQL_Running: | Yes |
| Replicate_Do_DB: | db_replikasi |
| 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: | 2082 |
| Relay_Log_Space: | 235 |
| 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 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 10. Proses aktivasi replikasi disisi slave.

### 3. KESIMPULAN

Berdasarkan dari penelitian yang telah dilaksanakan dan diuraikan pada penjelasan sebelumnya, maka model replikasi menggunakan *asynchronous* pada basis data terpusat dapat disimpulkan sebagai berikut:

1. Penelitian ini menghasilkan model arsitektur replikasi basis data yang dapat dipergunakan dalam proses pengiriman data.
2. Penerapan metode replikasi *asynchronous* menghasilkan sebuah terapan yang lebih baik dalam proses aliran data.

### Daftar Pustaka

- [1] Connolly, T. M., & Begg, C. E, Database Systems : A Practical Approach to Design, Implementation, and Management. (5th edition). Pearson Education.2010
- [2] Fathansyah, Sistem Basis Data Lanjutan, Informatika, Bandung.2004
- [3] Ladjamuddin. A.B, Konsep Sistem Basis Data Dan Implementasinya, Graha Ilmu Yogyakarta. 2004
- [4] Nurwenda.T.A, Aplikasi Pengolahan Data Kepegawaian Berbasis Web degan Database Terdistribusi menggunakan Replikasi Asynchronous PT. Taspen Kantor Utama Bandung. Unikom.2010.

### Biodata Penulis

**Ari Muzakir**, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Teknik Informatika Universitas Bina Darma Palembang, lulus tahun 2009. Memperoleh gelar *Master of Computer Science* (M.Cs) Program Pasca Sarjana Magister Ilmu Komputer Universitas Gajah Mada Yogyakarta, lulus tahun 2012. Saat ini menjadi Dosen di Universitas Bina Darma Palembang.