

## IMPLEMENTASI SISTEM TERDISTRIBUSI MENGGUNAKAN REPLIKASI DATABASE DAN WEB SERVICE

### IMPLEMENTATION OF A DISTRIBUTED SYSTEM USING DATABASE REPLICATION AND WEB SERVICE

Joko Triyono<sup>1\*</sup>, Putri Nadira<sup>2</sup>, Canu Ali Subkan<sup>3</sup>

<sup>1</sup>Jurusan Rekayasa Sistem Komputer, Fakultas Sains Terapan, Institut Sains dan Teknologi  
AKPRIND Yogyakarta

<sup>2,3</sup>Jurusan Informatika, Fakultas Teknologi Industri, Institut Sains dan Teknologi AKPRIND Yogyakarta

<sup>1\*</sup>jack@akprind.ac.id, <sup>2</sup>putrinadira@gmail.com, <sup>3</sup>canualisubhan12@gmail.com

\*penulis korespondensi

#### Abstrak

Perkembangan teknologi informasi dan komunikasi telah mempengaruhi segala aspek kehidupan, dan ini menyebabkan tingkat penggunaan teknologi menjadi sangat tinggi. Kemudahan dalam memperoleh dan mengelola informasi menjadi point penting untuk perusahaan dan atau instansi. Distribusi data dan aplikasi menjadi tantangan tersendiri karena informasi harus lebih mendekat kepada pemakai sehingga pemakai menjadi lebih fleksibel dalam mengelola informasi untuk menunjang kegiatan dan pekerjaan. Penggunaan replikasi database yang dikombinasikan dengan web service server menjadi salah satu alternatif agar pemakai kelompok transaksional dan manajerial bisa lebih leluasa dalam mengelola informasi. Dari hasil penelitian ini telah terbukti bahwa kombinasi yang tepat untuk implementasi aplikasi terdistribusi bisa dilakukan dan aman.

**Kata kunci : Database; Replikasi; Sistem Terdistribusi; Web Service**

#### Abstract

The development of information and communication technology has affected all aspects of life, and this causes the level of technology use to be very high. Ease of obtaining and managing information is an important point for companies and or agencies. Distribution of data and applications is a challenge because information must be closer to users so that users become more flexible in managing information to support activities and work. The use of database replication combined with a web service server is an alternative so that transactional and managerial group users can more freely manage information. From the results of this study it has been proven that the right combination for implementing distributed applications can be done and is safe.

**Keywords: Database; Replication; Distributed System; Web Service**

#### 1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi semakin hari semakin kompleks dan telah memberi pengaruh besar bagi segala aspek kehidupan, teknologi saat ini telah memberi banyak kemudahan dalam penyampaian suatu informasi. Informasi disimpan, diolah dan dihapus menjadi bagian utama dari ketersediaan informasi. Salah satu permasalahan penting yang dihadapi sebuah perusahaan atau instansi adalah tersedianya infrastruktur teknologi informasi yang memadai dalam mengelola dan melakukan pengamanan terhadap data penting, data pada umumnya menggunakan database untuk menyimpannya.

Database menjadi bagian yang tidak terpisahkan dan memiliki nilai tinggi. Ketersediaan data yang baik secara realtime terkadang menjadi kebutuhan yang rumit pada masalah tertentu. Dengan

sistem terpusat kadang menjadi kendala dalam beberapa hal, disebabkan mobilitas yang tinggi dari pemakai atau lokasi kantor yang berjauhan secara fisik dan geografis, sehingga sering mengganggu kinerja sistem. Sehingga muncul istilah sistem dan database terdistribusi

Proses replikasi data adalah sebuah proses yang sering dilakukan pada sistem database terdistribusi. Sistem database terdistribusi merupakan kumpulan dari node-node komputer. Setiap node tersebut dapat berpartisipasi dalam proses transaksi yang melakukan pengaksesan data pada sebuah aplikasi atau beberapa beberapa aplikasi. Terdapat beberapa hal yang dapat dijadikan alasan dalam membangun database terdistribusi, seperti pemakaian bersama (share), kehandalan (reliability), ketersediaan (availability) dan kecepatan pemrosesan query[2]. Keuntungan utama yang dapat kita peroleh dari database terdistribusi adalah kemampuan untuk pemakaian dan pengaksesan data secara bersamaan dengan handal serta efisien[3]. Replikasi berguna dalam hal peningkatan kinerja sistem serta dapat melindungi ketersediaan data yang akurat.

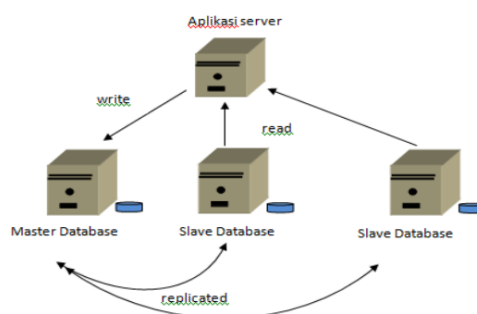
Pada penelitian terdahulu [4] tentang replikasi lebih menekankan kepada proses replikasi master slave, dengan proses transaksi tetap terjadi pada master, sedangkan pada penelitian saat ini proses transaksi bisa dilakukan secara terdistribusi pada aplikasi-aplikasi yang dekat dengan slave sedangkan hasil transaksi akan dikirimkan ke web master menggunakan web service.

Dalam penelitian ini akan diangkat tema Implementasi Sistem Terdistribusi menggunakan Replikasi Database dan Web Service dengan kasus pada simulasi sistem informasi akademik, dimana salah satu kegiatan akademik adalah proses krs, kegiatan lab dan kegiatan jurusan. Pada simulasi ini akan dilakukan distribusi sistem dari proses krs yang dilakukan secara terpusat dan akan di distribusikan ke unit-unit laboratorium dan jurusan-jurusan melalui sinkronisasi replikasi. Sedangkan jika unit-unit tersebut melakukan transaksi akan dilakukan melalui web service ke sistem informasi akademik terpusat.

## 2. DASAR TEORI /MATERIAL DAN METODOLOGI/PERANCANGAN

### 2.1 Sistem Terdistribusi

Sistem terdistribusi berbeda dengan klien server, di mana server terhubung ke satu server lain dalam jaringan, dan jika terjadi crash pada salah satu server, data yang di akses akan diambilkan dari server lain. Sistem terdistribusi terlihat pada gambar 1 adalah sistem di mana komponen-komponen dalam sistem informasi didistribusikan ke berbagai lokasi di jaringan komputer [1].

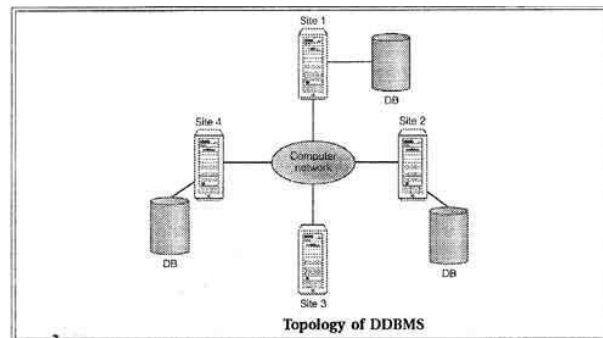


Gambar 1. Sistem Terdistribusi

### 2.2 Database Terdistribusi

Database terdistribusi pada gambar 2 pada dasarnya digunakan untuk menyimpan data dalam jumlah yang sangat besar pada situs yang berbeda atau server. Ada berbagai partisi teknik menggunakan database terdistribusi, seperti vertikal partisi. Masa depan bekerja terutama

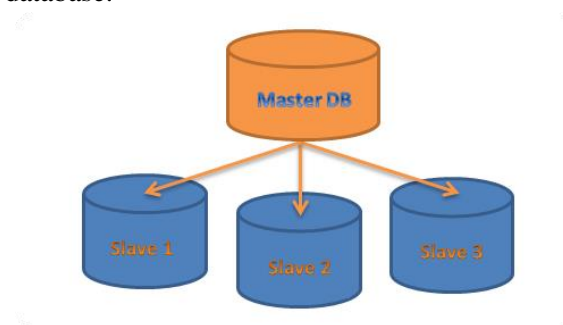
berkonsentrasi pada penggunaan terdistribusi data pertambahan di database terdistribusi [2]. Persyaratan untuk peningkatan sistem manajemen database terdistribusi menjadi lebih penting, system terdistribusi bervariasi dari system terpusat dalam satu data dan penguasaan data menyebar lebih dari dua atau lebih secara fisik memisahkan lokasi. System manajemen database terdistribusi adalah pusat untuk banyak ancaman keamanan tambahan kepada mereka yang hadir dalam system manajemen database terpusat. Selain itu perluasan keamanan database terdistribusi yang memadai telah sulit dengan pembukaan relatif baru model berorientasi objek database. Replika baru ini tidak dapat diabaikan. Telah dibuat untuk mengatasi kesulitan tubuh data yang disimpan dalam system database yang hadir [3].



Gambar 2 Database Terdistribusi

### 2.3 Replikasi

Replikasi adalah pengoperasian menyimpan bagian dari database, sebagai salinan, pada beberapa node pada sebuah jaringan. Jika penggunaan update salinan local kemudian database update secara otomatis semua salinan data dengan asumsi replica saling konsisten, replikasi meningkatkan ketersediaan karena transaksi dapat membaca salah satu copies. Selain itu, replikasi menyediakan keandalan yang lebih meminimal kan kemungkinan kehilangan total data, dan sangat meningkatkan pemulihan bencana. Meskipun replikasi memberikan kepada system performa yang lebih baik membaca itu mempengaruhi system negatif ketika salinan database yang dimodifikasi misalnya harus diterapkan ke semua salinan untuk menjaga konsistensi saling direplikasi [4] gambar 3 menunjukkan replikasi database.

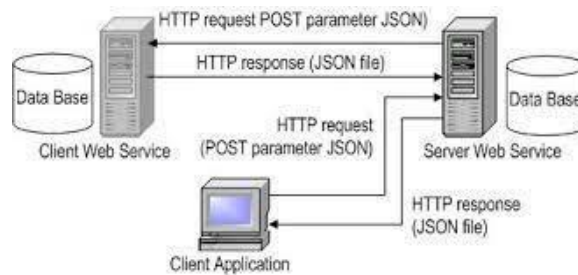


Gambar 3 Replikasi Database

### 2.4 Web Service

Web service merupakan aplikasi yang berisi sekumpulan basis data (*database*) dan perangkat lunak (*software*) atau bagian dari program perangkat lunak yang diakses secara *remote* oleh piranti dengan perantara tertentu. Melalui web service, memungkinkan pengguna untuk mengatasi permasalahan berupa *interoperability* dan mengintegrasikan sistem berbeda.

Pada umumnya, web service memiliki ciri khusus berupa URL layaknya web. Yang membuat berbeda adalah interaksi yang diberikan oleh web service itu sendiri. URL pada web service hanya mengandung sekumpulan informasi, perintah, dan konfigurasi (sintaks yang berguna untuk membangun fungsi tertentu dari aplikasi). Web service mampu menukar data tanpa memandang sumber *database*, bahasa yang digunakan, dan pada platform apa data tersebut dikonsumsi. Kemampuan itulah yang memungkinkan web service menjadi jembatan penghubung untuk berbagai sistem [5].

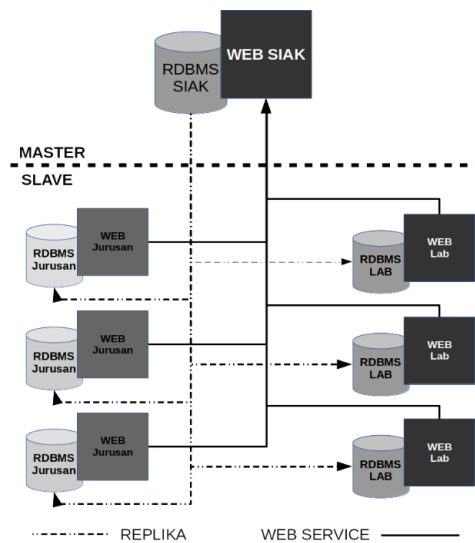


Gambar 4 Web Service

## 2.5 Metodologi

Tahapan penelitian ini dibagi kedalam tiga tahap, yaitu perancangan dan arsitektur sistem, identifikasi kebutuhan sistem dan implementasi sistem.

### 2.5.1 Perancangan Arsitektur Sistem



Gambar 5 Arsitektur Sistem

Arsitektur sistem yang akan dibuat adalah pertama (1) dengan membangun database yaitu master (siak), slave (Jurusan dan Laboratorium). Dan kedua (2) Membangun Aplikasi Terdistribusi yaitu Web Service Server Akademik, Web Service Client Jurusan dan Web Service Client Laboratorium seperti terlihat pada gambar 5.

Dari gambar 5 terlihat bahwa dari RDBMS SIAK data akan di replikasi ke semua RDBMS Jurusan dan Lab, sedangkan jika terjadi transaksi di web jurusan maupun web lab, data akan dikirimkan melalui web service ke Web SIAK. Sehingga perubahan data hanya akan terjadi di rdbms SIAK saja, sedangkan semua RDBMS di slave akan menerima hasil sinkronisasi master slave.

### 2.5.2 Identifikasi Kebutuhan Sistem

Menyiapkan Master Slave Database, setelah disiapkan database di SIAK, maka Langkah yang harus dilakukan adalah membuat master slave database.

#### Master Database

1. Menentukan Identitas Server (harus unik)

Pada file my.cnf pada linux atau my.ini pada windows

```
[mariadb]
log-bin
server_id=111
log-basename=siak
binlog-format=mixed
```

2. Membuat User Replikasi

Membuat user pada posisi root

```
CREATE USER 'replika_siak'@'%' IDENTIFIED BY 'rahasia';
GRANT REPLICATION SLAVE ON *.* TO 'replika_siak'@'%';
```

Untuk melihat status master maka gunakan perintah : SHOW MASTER STATUS;

#### Slave Database

1. Menentukan Identitas Server Slave (harus Unik)

Untuk masing-masing database slave dengan server\_id 200 dst

```
[mariadb]
log-bin
server_id=200
log-basename=siak
binlog-format=mixed
binlog-do-db=siak
```

2. Memulai Slave

Untuk bisa menjalankan slave, maka harus ditentukan dulu master yang dituju oleh slave menggunakan change master to,

```
CHANGE MASTER TO
MASTER_HOST='192.168.0.108',
MASTER_USER='replika_siak',
MASTER_PASSWORD='rahasia';
```

Setelah itu barulah di jalankan perintah

START SLAVE;

Untuk melihat status slave maka gunakan perintah : SHOW SLAVE STATUS;

Menyiapkan Web Service Server dan Client

### 2.5.3 Implementasi Sistem Web Service Server dan Client

Untuk implementasi web service server dan client, maka harus ditentukan dulu aplikasi transaksi yang akan dilakukan. Dalam pengujian ini, transaksi yang akan dilakukan oleh client adalah mengisi nilai sebuah matakuliah praktikum. Sehingga Client akan melakukan transaksi pengisian nilai praktikum dan saat disimpan client akan mengirimkan data tersebut ke Web Service Server, yang oleh web service server akan ditransaksikan ke database master.

### 2.5.4 Metode Pengujian Sistem

Pengujian Sistem dilakukan dengan menguji Master Slave apakah bisa berjalan dengan normal, selanjutnya dilakukan pengujian transaksional menggunakan aplikasi tersebar dengan mengirimkan transaksi melalui web service ke web server master. Pengamatan dilakukan dengan memperhatikan log transaksi baik di master maupun slave dan juga database pada keduanya.

## 3. PEMBAHASAN

Dalam pengujian ini dilakukan simulasi skala laboratorium, dimana Master terletak pada ip 192.168.0.108, Slave Laboratorium pada ip 192.168.0.111 dan Slave Jurusan pada ip 192.168.0.112

### 3.1 Master

Master pada ip 192.168.0.108, setelah di setting pada file my.ini akan terlihat statusnya sebagai berikut:

```
MariaDB [(none)]> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| siak-bin.000007 |      633 | siak         |                   |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

MariaDB [(none)]>

Terlihat pada status master dengan file log *siak-bin.000007* dengan binlog\_do\_DB *siak*.

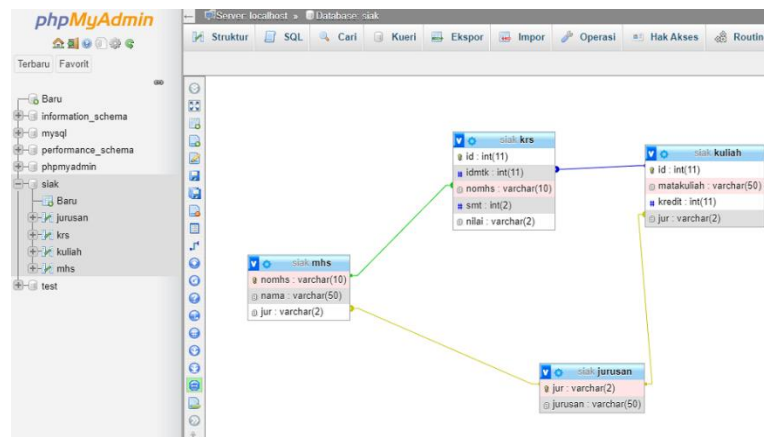
### 3.2 Slave

Slave laboratorium dan jurusan, akan terlihat statusnya sebagai berikut:

Terlihat *Slave\_IO\_State* : *Waiting for master to sent event*, dengan *Master\_Host*: 192.168.0.108, *Master\_User*: *replica\_siak*, *Master\_Port*: 3306, *Master\_Log\_File*: *siak-bin.000007*. Dengan status *Slave\_IO\_Running*: *Yes*, *Slave\_SQL\_Running*: *Yes*. Dari kondisi tersebut maka bisa dinyatakan bahwa slave telah berjalan dengan sempurna.

### 3.3 Design Database

Saat bekerja dengan modus master slave, maka apa yang dilakukan pada master tentunya akan terkirim / ter-replikasi ke slave. Gambar 6 menunjukkan design dari databasae siak.



Gambar 6 Design Database

### 3.4 Aplikasi Jurusan

Pada saat aplikasi jurusan melakukan update data, data yang dibaca adalah data yang ada di slave jurusan, sedangkan proses update data akan dikirimkan ke web service server siak. Gambar 7 menunjukkan saat aplikasi jurusan menambahkan data matakuliah beserta script pada sisi jurusan dan web service server, Gambar 8 adalah posisi tabel kuliah pada slave jurusan dan master. Setelah eksekusi selesai akan terlihat posisi tabel kuliah, logbin pada master dan slave pada gambar 10.

Pemrograman Komputer  
3  
07  
Add MTK  
FORM ISI MATAKULIAH

```
<?php
$datain="".$_GET['mtk'].''&''.$_GET['kredit'].''&''.$_GET['jur'].
'';
$kunci=base64_encode($datain);
$file=file_get_contents('http://192.168.0.108/siak/admatakuliah.php?q
='.$kunci);
?>
```

Script aplikasi jurusan

```
<?php
//menambah-matakuliah
include("buka.php");
$datain=base64_decode($_GET['q']);
$x=explode("&",$datain);
$sql="insert into kuliah(matakuliah, kredit, iur)
values('".$x[0]."', '".$x[1]."', '".$x[2]."');
if($conn->query($sql)==TRUE) echo "Sukses "; else echo "gagal";
$conn->close();
?>
```

Script aplikasi web service server

Gambar 7 Aplikasi Tambah matakuliah dan Script

```
MariaDB [(none)]> use siak
Database changed
MariaDB [siak]> select * from kuliah;
```

id	matakuliah	kredit	jur
1	Pemrograman Klient Server	3	07
2	Pemrograman Antar Platform	3	05
3	pancasila	2	05

3 rows in set (0.000 sec)

**MASTER** ←

```
MariaDB [(none)]> use siak
Database changed
MariaDB [siak]> select * from kuliah;
```

id	matakuliah	kredit	jur
1	Pemrograman Klient Server	3	07
2	Pemrograman Antar Platform	3	05
3	pancasila	2	05

3 rows in set (0.107 sec)

→ **SLAVE**

Gambar 8 Posisi Tabel Kuliah sebelum transaksi pada Slave dan Master

Dari log master pada gambar 10 pengujian terlihat bahwa terjadi perintah *query insert ...* pada (12. row) posisi log 1077 dan end log 1227. Selanjutnya pada (13. Row) terlihat proses commit pada posisi log 1227 dan end log 1258 dan data sudah bertambah pada id 4. Sedangkan pada slave seperti pada gambar 11 terlihat data sudah bertambah pada id 4 dan posisi *read\_master\_log\_pos* pada posisi 1258. Kondisi ini menunjukkan bahwa proses replikasi sudah berjalan dengan baik.

```

***** 12. row *****
Log_name: siak-bin.000008
Pos: 1077
Event_type: Query
Server_id: 111
nd_log_pos: 1227
Info: use 'siak'; insert into kuliah(matakuliah, kredit, jur) values('Pemrograman Komputer','3','07')
***** 13. row *****
Log_name: siak-bin.000008
Pos: 1227
Event_type: Xid
Server_id: 111
nd_log_pos: 1258
Info: COMMIT /* xid=229 */
(3 rows in set (0.000 sec))

MariaDB [siak]> select * from kuliah;
+----+-----+-----+-----+
| id | matakuliah | kredit | jur |
+----+-----+-----+-----+
| 1 | Pemrograman Klient Server | 3 | 07 |
| 2 | Pemrograman Antar Platform | 3 | 05 |
| 3 | pancasila | 2 | 05 |
| 4 | Pemrograman Komputer | 3 | 07 |
+----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [siak]> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 192.168.0.108
Master_User: replika_siak
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: siak-bin.000008
Read_Master_Log_Pos: 1258
Relay_Log_File: mysql-relay-bin.000015
Relay_Log_Pos: 1556
Relay_Master_Log_File: siak-bin.000008
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:

```

Gambar 9 Posisi tabel kuliah dan logbin master dan slave setelah transaksi

## 4. KESIMPULAN

### 4.1 Kesimpulan

Dari hasil penelitian ini dapat disimpulkan bahwa proses replikasi yang dikombinasikan dengan web service bisa berjalan dengan sempurna, sehingga bisa diwujudkan aplikasi terdistribusi yang lebih memberikan kenyamanan kepada pemakai.

### 4.2 Rekomendasi

Adapun saran untuk pengembangan penelitian selanjutnya adalah harus diatur secara ketat proses komunikasi data, format data transaksi dan penjadwalan sinkronisasi dari master ke slave.

## DAFTAR PUSTAKA

- [1] A. P. Sembiring, "The Replication of Academic Database System using Linux Ubuntu," *Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE)*, pp. 70-76, 2020.
- [2] E. Sutanta and A. Ashari, "Distribusi Basis Data Kependudukan untuk Optimalisasi Akses Data: Suatu kajian pustaka," *Jurnal Ilmu Komputer*, vol. 5(1), pp. 1-9, 2012.
- [3] A. Rauf, "Aplikasi Database Terdistribusi pada KSP Sinar Kasih Batam," *SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, vol. Vol 8 No 2, 2019.
- [4] J. Triyono, "Replikasi Untuk Meningkatkan Kinerja dan Ketersediaan Data (Study Kasus Sistem Informasi Akademik).," *Jurnal Technoscientia*, vol. Vol 5 No 1, pp. 31-40, 2012.
- [5] Dicoding, "Apa itu Web Service? Beserta Pengertian dan Contohnya," Dicoding, 13 januari 2021. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-web-service/>. [Accessed 8 12 2021].