

MEMBANGUN SERVER DAN ANALISIS BACKUP DATABASE POSTGRESQL MENGGUNAKAN TEKNIK REPLICATION MASTER/SLAVE

Zaenal Arifin¹, Joko Triyono², Rr Yuliana Rachmawati³

Program Studi Teknik Informatika, Fakultas Teknologi Industri

Institut Sains & Teknologi AKPRIND Yogyakarta

Email : ¹ email.milzam@gmail.com, ² jack@akprind.ac.id, ³ yuliana@akprind.ac.id

ABSTRACT

A database is a collection of information stored on a computer systematically so that it can be checked using a program to obtain information from the database. PostgreSQL or often called Postgres is one large database that offers scalability, flexibility, and high performance. Replication is the process of copying data from a database on one master server to a slave database server. Database replication can be applied to the PostgreSQL database server to backup the main database.

In this study, building 2 PostgreSQL database servers or what is often called master / slave database replication, the replication method used is synchronous to replicate the database in real time. This study discusses the analysis of the accuracy of replication data and the delay of database replication time. The material used for testing is Student data to determine the accuracy of the data between the master server and the slave server. The data collection methodology is carried out through observation and literature. The method in database replication analysis testing uses benchmarking.

The discussion of this research by testing uses parameter transactions and client requests. Testing two master servers and slave servers that have been configured by the PostgreSQL database by replicating the Student database using the synchronous replication method, the master server will replicate in real-time to the slave server. Furthermore, the results of the replication test analysis produce the accuracy of the Student data between the master and slave and the delay time needed for database replication produces time with an average of 0 / ms.

Keywords: PostgreSQL, Replication Master/Slave, Database, delay, backup.

INTISARI

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program untuk memperoleh informasi dari basis data tersebut. PostgreSQL atau sering disebut Postgres merupakan salah satu database besar yang menawarkan skalabilitas, keluwesan, dan kinerja tinggi. PostgreSQL termasuk dalam kategori *free open source software* (FOSS). Penggunaannya begitu meluas diberbagai *platform* dan didukung oleh banyak bahasa pemrograman.

Dalam penelitian ini merancang server *backup* database PostgreSQL atau yang sering disebut replikasi database *master/slave*. Replikasi adalah proses menyalin data dari database pada satu server (*Master*) ke database server lain (*Slave*). Tujuan dari penelitian ini adalah untuk mendistribusikan database ke beberapa mesin, jadi ketika server master memiliki masalah, ada mesin cadangan (*slave*) dengan yang sama tersedia untuk menangani masalah yang terjadi pada server *master*.

Analisis dalam penelitian ini bertujuan untuk mengetahui keakuratan data replikasi database antara server *master* dan server *slave* dan untuk mencari berapa lama waktu (*delay*) yang dibutuhkan data tereplikasi ke server *slave* dan hasil analisis didapatkan dari log yang ada di kedua server tersebut. Hasil penelitian ini pengujian replikasi database *master/slave* server PostgreSQL berjalan dengan baik.

Kata kunci: PostgreSQL, Replication Master/Slave, Database, delay, backup.

PENDAHULUAN

Menghadapi banyaknya permintaan kebutuhan informasi pada lingkungan global seperti sekarang ini, diperlukan analisa terhadap kebutuhan dan high availability (ketersediaan tinggi) yang dimiliki untuk menangani banyaknya permintaan tersebut. Perusahaan tidak akan mampu kehilangan pelanggannya ketika layanan yang ditawarkan menjadi terbatas terutama karena downtime server yang hampir pasti akan terjadi dan mengakibatkan kerugian dengan biaya yang tinggi.

Faktanya server tidak berjalan selamanya. Permasalahan dapat terjadi karena adanya kerusakan pada komponen-komponen hardware (perangkat keras) di komputer server, software (perangkat lunak) yang menyebabkan crash (kerusakan), atau pun sistem operasi pada komputer server yang perlu shutdown (dimatikan) untuk melakukan upgrade dan proses maintenance (pemeliharaan). Oleh karena itu dibutuhkan teknologi untuk menangani backup data sehingga data tersebut dapat digunakan kembali apabila terjadi kesalahan atau kerusakan pada server utama (Imam, 2017).

Pendekatan umum untuk meningkatkan ketersediaan sistem adalah dengan menggunakan metode replikasi synchronous adalah proses penyimpanan data pada lebih dari satu server disaat yang bersamaan, yang pada umumnya terdiri dari master dan slave server. Proses pengolahan data yang meliputi perubahan, penambahan, atau pengurangan data pada master server, juga dilakukan secara langsung di slave server. Manfaat dari replikasi adalah mendistribusikan database ke beberapa mesin, jadi ketika server master memiliki masalah, ada server slave atau backup dengan data yang sama tersedia untuk menangani masalah yang terjadi di server master.

Berdasarkan beberapa hal yang dijelaskan, penulis akan melakukan penelitian replikasi database dengan membangun server master untuk server database utama dan server slave yang akan dijadikan sebagai server backup. Setelah dilakukan pengujian replikasi master/slave penelitian melakukan analisis terhadap hasil pengujian untuk menghasilkan replikasi yang akurat.

TINJAUAN PUSTAKA

Penelitian ini menggunakan pustaka hasil-hasil penelitian sebelumnya yang relevan. Referensi yang diambil berupa jurnal yang berhubungan dengan penelitian ini.

Tinjauan pustakan yang pertama diambil dari jurnal penelitian (Triyono, 2012) berjudul Replikasi Untuk Meningkatkan Kinerja dan Ketersediaan Data Pada Sistem Informasi Akademik. Penelitian ini membahas tentang pencarian algoritma-algoritma untuk merancang arsitektur database terdistribusi dengan metode *replikasi* yang diimplementasikan pada database akademik .

Tinjauan pustaka yang kedua diambil dari jurnal penelitian (Dolly, 2015) dengan judul Pengembangan Sistem Replikasi dan Redudansi Untuk Meningkatkan Keandalan Basisdata MySQL. Tujuan dari penelitian membangun sebuah rancangan virtualisasi aplikasi server master slave basisdata yang terhubung atau terkoneksi dengan manajemen node dan slave basisdata, menggunakan aplikasi Virtualbox untuk mendapatkan hasil performa sistem replikasi dan redudansi dalam merancang sistem cluster. Basisdata kluster yang diimplementasikan dengan mesin kluster di sini bersifat *high-availability*.

Tinjauan pustaka yang ketiga diambil dari jurnal penelitian (Imam, 2017) pada penelitian ini sebuah penanganan ketika gangguan *downtime server* dengan menggunakan teknologi *high availability cluster* dengan metode *synchronous replication*, *redundancy*, dan *failover*.

Berdasarkan beberapa penelitian-penelitian di atas disimpulkan bahwa replikasi database mampu menangani masalah backup data untuk mengantisipasi adanya kegagalan di server master. Dengan replikasi database menggunakan metode synchronous, server master mampu melakukan transaksi dan kemudian direplikasi ke server slave secara *real-time*.

Metodologi penelitian

Untuk mengetahui jalanya implementasi dan analisis data yang tereplikasi dari server *master* ke server *slave*. Di sini ada beberapa langkah, yaitu:

1. Langkah pertama mulai awal alir langkah penelitian.
2. Analisis Kebutuhan

Analisis kebutuhan merupakan tahap melakukan pengumpulan data dari *software* dan *hardware* jaringan yang akan digunakan dalam analisis penelitian. Pengumpulan dapat dilakukan dengan cara mencari studi pustaka dan observasi. Setelah analisis kebutuhan sistem dilakukan, selanjutnya masuk ke tahap perancangan topologi jaringan.

3. Perancangan Topologi Jaringan

Tahap perancangan topologi jaringan merupakan tahap rancangan jaringan yang akan digunakan antara server *master*, server *slave*, dan *client* untuk *remote*.

4. Membangun Mesin *Virtual Private Server* (VPS)

Membangun sebuah mesin server *virtualisasi* untuk kedua membangun kedua server *master* dan *slave*. Mesin *virtual* ini dibuat menggunakan *Google Cloud Platform*.

5. Instalasi *Operating System* Server *Master* dan Server *Slave*

Pada tahap ini membangun sebuah server, sistem operasi yang digunakan adalah *Linux Ubuntu 14.04* dinstal pada sebuah mesin *virtual private server* (VPS) yang sudah disediakan.

6. Konfigurasi Server

Pada tahap ini mengkonfigurasi di masing-masing server untuk menggunakan teknik replikasi. Adapun langkah-langkah yang harus dilakukan diantaranya menghubungkan antara server *master* dengan server *slave/backup* agar dapat bekerja sama. Setelah itu mengkonfigurasi database *PostgreSQL* pada kedua server baik server utama maupun server *backup*.

7. Replikasi Database *PostgreSQL*

Pada tahap ini menggunakan teknik *replication* untuk *backup* database *PostgreSQL*. mereplikasi data yang ada pada database server utama ke database *backup*. Metode replikasi bertujuan untuk menyinkronkan database dari *PostgreSQL* server, untuk terciptanya suatu *multiple server*.

8. Pengujian Database *PostgreSQL*

Melakukan pengujian replikasi antara database *PostgreSQL* server apakah tereplika dengan tepat.

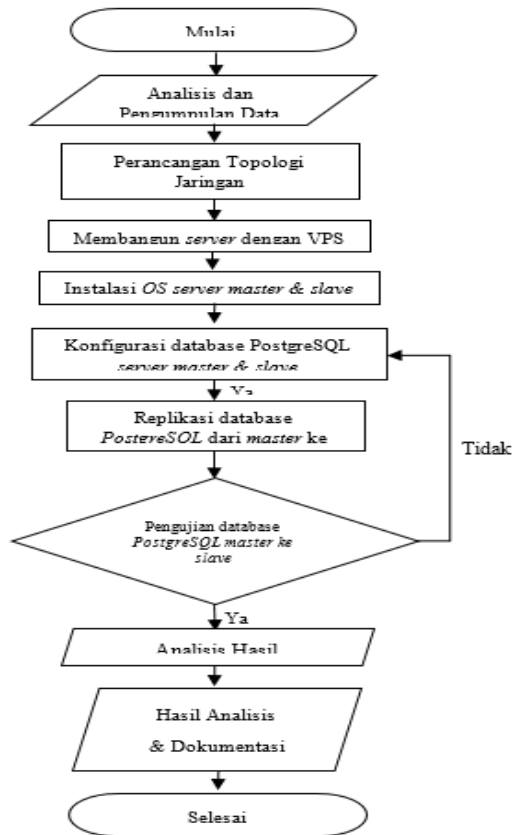
9. Analisis Hasil Pengujian

Setelah melakukan pengujian replikasi database, setelah itu melakukan analisa database *PostgreSQL* dengan mencatat *record* aktivitas dari *log PostgreSQL* untuk mencatat keakuratan replikasi data dan mencatat seberapa cepat waktu rata-rata replikasi database dari server utama ke server *backup*.

10. Pengambilan Data dan Kesimpulan

Mengambil kesimpulan dari hasil analisis repliasi database *PostgreSQL* dan penelitian yang telah dilakukan.

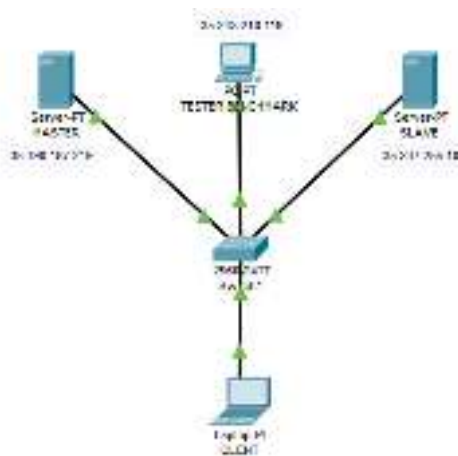
Diagram alir langkah penelitian implementasi replikasi database *PostgreSQL* ditunjukkan pada gambar :



Gambar 1. Diagram Alir Penelitian

PEMBAHASAN

Teknik replikasi database membutuhkan database dan akses sumber data lebih dari 1 komputer server, di mana terdapat 1 buah kompute server yang bertidak sebagai Master, dan terdapat satu buah komputer yang bertidak sebagai server Slave. Berikut topologi yang akan digunakan :



Gambar 2. Topologi Jaringan

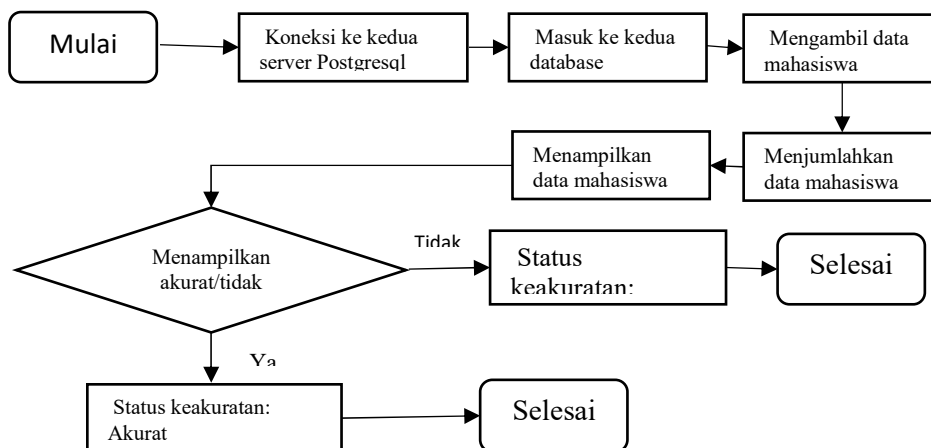
Pada gambar 2 adalah sebuah perancangan jaringan menggunakan topologi start dengan membangun 1 host VPS (Master) dengan IP Public 35.190.187.219, 1 host VPS (Slave) digunakan sebagai server slave PostgreSQL dengan IP Public 35.237.255.10, 1 host VPS dengan IP Public 35.243.210.116 digunakan sebagai server tester untuk pengujian pada server dengan menggunakan benchmark dan satu computer untuk remote ketiga server menggunakan SSH atau menjadi client.

```

<?php
$HostName = 'host = 35.190.187.219';
$port = "port = 5432";
$dbname = 'dbname = mahasiswa';
$credentials = 'user = postgres password = 1234';
$file = new SplFileObject("insert_50_mhs.txt");
while (!$file->eof()) {
    // $connection = pg_connect("". $HostName. " ". $port. " ". $dbname. " ". $credentials);
    $connection = pg_connect("host = '35.190.187.219' port = '5432' dbname = 'mahasiswa' user = 'postgres' password = '1234'");
    if (!$connection)
    {
        echo "Error : Unable to open database\n";
    } else {
        echo "Opened database successfully\n";
        $explode = array();
        $explode = explode(" ", $file->fgets());
        if ($explode[0] != '') {
            $sql = "INSERT INTO mahasiswa (nim, nama, jenjang, kampus, prodi)
VALUES ('$explode[0]', '$explode[1]', '$explode[2]', '$explode[3]', '$explode[4]')";
            if (pg_query($connection, $sql))
            {
                echo "New record ". $sql. " has been created <br>";
            }
            else
            {
                echo "Error: " . $sql. "<br>". $connection->error;
            }
        }
    }
}
    
```

Gambar 3. Script input data

Script ini bertujuan untuk mempermudah input data dalam jumlah besar sekaligus. Dengan script ini input tidak dengan satu persatu, hal ini untuk memperoleh data yang maksimal. Setelah data yang di input menggunakan script selanjutnya data akan ditampilkan hasil data yang di input. Setelah data di input, untuk mengetahui data udah masuk atau tidak hasilnya akan dilihat di browser.



Gambar 4. Alur keakuratan data

Gambar 4 merupakan diagram alur untuk mengambil keakuratan data dengan mengkoneksikan ke server database master server dan database server slave. Kemudian mengeksekusi data mahasiswa, untuk dijumlahkan keseluruhan anggota yang telah diinput. Setelah data di jumlahkan, data akan ditampilkan dan mengecek kesamaan data dari nama dan NIM mahasiswa.

```

zaenz19@tester: ~
zaenz19@tester:~$ ab -n 10000 -c 50 http://35.237.255.10:5432/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 35.237.255.10 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

Server Software:
Server Hostname:      35.237.255.10
Server Port:          5432

Document Path:        /
Document Length:      0 bytes

Concurrency Level:    50
Time taken for tests:  11.321 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    0 bytes
HTML transferred:     0 bytes
Requests per second:  883.34 [#/sec] (mean)
Time per request:     56.604 [ms] (mean)
Time per request:     1.132 [ms] (mean, across all concurrent requests)
Transfer rate:         0.00 [Kbytes/sec] received
    
```

Gambar 5. Benchmark dengan parameter 10000 transaksi & 50 request

Pada gambar 5 adalah hasil pengujian yang dilakukan benchmark terlihat request per second dari server slave yaitu mampu melayani 883.34/sec untuk client yang dituju ke server slave. Berikut data hasil real yang diperoleh pada saat pengujian.

Total Mahasiswa Database Master :400		Total Mahasiswa Database Slave :400		
NIM Mahasiswa DB Master	Nama DB Master	NIM Mahasiswa DB Slave	Nama DB Slave	Akurat / Tidak ?
1323716	ACHMAD MUSTAIN	1323716	ACHMAD MUSTAIN	Akurat
161321013	ACHMAD ODYK AKBAR NAGARA	161321013	ACHMAD ODYK AKBAR NAGARA	Akurat
2323810	ADE SUPRIATNA	2323810	ADE SUPRIATNA	Akurat
1323718	ADHIN YULI PRASETYO	1323718	ADHIN YULI PRASETYO	Akurat
1323761	ADI PURWANTO	1323761	ADI PURWANTO	Akurat
181321006	AFAN TULUS SATRIA	181321006	AFAN TULUS SATRIA	Akurat
161321018	AFIF WIDIANTO	161321018	AFIF WIDIANTO	Akurat
141321007	AGIL PRABOWO	141321007	AGIL PRABOWO	Akurat
151321005	AGUNG HARWANTO	151321005	AGUNG HARWANTO	Akurat
1323736	AGUNG HARI PRASETYA	1323736	AGUNG HARI PRASETYA	Akurat

Gambar 6. Hasil rael/ keakuratan data

Gambar 6 merupakan real hasil keakuratan data yang diinput bersamaan dengan pengujian metode benchmark terhadap server slave. Berikut tabel hasil pengujian keakuratan replikasi database slave dan delay waktu replikasi, dengan parameter 10000 transaksi dan 50 request client.

Tabel 1. Benchmark 10000 transaksi & 50 request cilent

Jumlah Data Input	Total Database Terakhir		Status Keakuratan	Waktu log sql		Delay Waktu (Detik)
	DB Master	DB Slave		DB Master	DB Slave	
50	650	650	Akurat	00:48:14	00:48:14	0

Pada pengujian ini menggunakan benchmark pada server slave dengan menggunakan parameter uji coba 10000 transaksi dan 50 request client dengan input ke database mahasis di server master. Dengan pengujian tersebut server slave masih berjalan lancar tanpa ada delay dengan selisih waktu 0 ketika data direplikasi dilihat pada tabel 1.

2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	2	INSERT	2018-12-09	Mar-00	0	LOG	0	duration: 3.589 ms
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	3	idle	2018-12-09	Mar-39	0	LOG	0	statemen
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	4	INSERT	2018-12-09	Mar-00	0	LOG	0	duration: 2.149 ms
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	5	idle	2018-12-09	Mar-40	0	LOG	0	statemen
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	6	INSERT	2018-12-09	Mar-00	0	LOG	0	duration: 2.410 ms
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	7	idle	2018-12-09	Mar-41	0	LOG	0	statemen
2018-12-09 00:48:14	postgres	mahasisw	4169	36.73.43.61	5c0c664c	8	INSERT	2018-12-09	Mar-00	0	LOG	0	duration: 1.959 ms
2018-12-09 00:48:15	postgres	mahasisw	4169	36.73.43.61	5c0c664c	9	idle	2018-12-09	Mar-42	0	LOG	0	statemen
2018-12-09 00:48:15	postgres	mahasisw	4169	36.73.43.61	5c0c664c	10	INSERT	2018-12-09	Mar-00	0	LOG	0	duration: 1.977 ms
2018-12-09 00:48:15	postgres	mahasisw	4169	36.73.43.61	5c0c664c	11	idle	2018-12-09	Mar-43	0	LOG	0	statemen

Gambar 7. Pg_log transaksi input 50 server master

2018-12-09 00:48:14	28016	35.243.210.11	5c0c664e.6d70	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28022	35.243.210.11	5c0c664e.6d76	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28023	35.243.210.11	5c0c664e.6d77	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28024	35.243.210.11	5c0c664e.6d78	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28025	35.243.210.11	5c0c664e.6d79	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28026	35.243.210.11	5c0c664e.6d7a	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28015	35.243.210.11	5c0c664e.6d6f	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28027	35.243.210.11	5c0c664e.6d7b	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28028	35.243.210.11	5c0c664e.6d7c	1	2018-12-09	0	LOG	08P01	invalid length of startup packet
2018-12-09 00:48:14	28029	35.243.210.11	5c0c664e.6d7d	1	2018-12-09	0	LOG	08P01	invalid length of startup packet

Gambar 8. Pg_log transaksi input 50 server slave

Dilihat pada gambar 7 dan gambar 8 log transaction kedua server master dan slave ketika terjadi pengujian keakuratan data dan delay replikasi.

Pengujian

Setelah dilakukan pengujian dengan metode *benchmark* server dengan parameter transaction 10000 dan *request client* 50, 100, 250, 500 terhadap server *slave* untuk melayani *request client* dan saat pengujian tersebut dilakukan *input* data ke database server *master*. Analisis terhadap sistem replikasi database PostgreSQL *master/slave* akan diperoleh hasil keakuratan data replikasi dan *delay* waktu yang dibutuhkan untuk *replication*. Berikut tabel hasil seluruh pengujian tersebut:

Tabel 2. Hasil analisis semua pengujian

Jumlah Data Input	Benchmark		Total Database Terakhir		Status Keakuratan	Waktu pg_log		Delay Waktu (Detik)
	Transaksi	Request	DB Master	DB Slave		DB Master	DB Slave	
50	10000	50	650	650	Akurat	00:48:14	00:48:14	0
100	10000	100	750	750	Akurat	01:36:49	01:36:49	0
250	10000	250	1000	1000	Akurat	03:43:44	03:43:44	1

Tabel di atas adalah hasil dari pengujian yang dilakukan untuk replikasi data PostgreSQL, berdasarkan analisis dapat menghasilkan keakuratan data replikasi dan waktu delay yang dibutuhkan untuk replikasi dari server master ke server slave. Berikut penjelasan dari tabel hasil pengujian:

- a. Jumlah data yang diinput dalam setiap pengujian adalah 50, 100, dan 250.

- b. Pengujian menggunakan metode benchmark dengan parameter transaksi 10000 dan request client dengan jumlah 50, 100, dan 250 yang dituju ke server slave.
- c. Total database terakhir di dua server master dan slave catatan data yang masuk setelah dilakukan input pengujian. Data yang diinput ke server master jumlahnya akan sama dengan database di server slave.
- d. Status keakuratan dalam pengujian untuk memastikan data yang masuk antara database server master dan server slave akurat atau tidak.
- e. Pg_log adalah catatan waktu yang ada pada setiap server ketika terjadi transaksi.
- f. Delay waktu diambil dari pg_log yang ada di setiap server dengan sftp, dari log waktu server master dikurangi log server slave untuk mengetahui delay waktu.

KESIMPULAN

Setelah melakukan membangun 2 server replikasi database *master/slave server* PostgreSQL dengan metode replikasi *synchronous* dan dilakukan pengujian replikasi database dengan metode *benchmark* dengan parameter 10000 transaksi dan 50, 100, 250 tertentu server *master* sudah mampu melakukan replikasi database ke server *slave* secara *real-time* dengan memiliki keakuratan data 100% dan mempunyai waktu yang dibutuhkan untuk replikasi database atau waktu *delay* rata-rata=0.

DAFTAR PUSTAKA

- Abidin, R. (2016). Pengertian Location Based Services (LBS) dan Komponennya. Retrieved August 30, 2017, from <https://teknojurnal.com/pengertian-location-based-services-lbs-dan-komponennya/>
- Ansari, M. A., Sharma, S., Shete, A., Ghorpade, S., & Ghodke, N. (2017). Event Organization Using GPS Based Location Tracking Including Communication System. *Imperial Journal of Interdisciplinary Research*, 3(6), 245–249.
- Godam. (2012). ARTI PENGERTIAN FASILITAS UMUM DAN FASILITAS SOSIAL (PERBEDAAN FASUM FASOS). Retrieved August 30, 2017, from <http://www.organisasi.org/1970/01/arti-pengertian-fasilitas-umum-dan-fasilitas-sosial-perbedaan-fasum-fasos.html#>
- Permatasari, D., & Noveristi, N. (2013). Sistem Informasi Pelayanan Pengaduan dan Tagihan Pelanggan di PDAM Tirtawening Kota Bandung Menggunakan SMS Gateway. *JAMIKA Jurnal Manajemen Informatika UNIKOM*, 3.
- Putra, E. B., Ichsan, D. M., & Elidjen. (2013). *APLIKASI DAN PERANCANGAN SISTEM PENGADUAN MASYARAKAT TERHADAP PEMKAB MERANGIN BERBASIS WEB*. Jakarta.
- Ramchandra Nimbalkar, R. (2013). Mobile Application Testing and Challenges. *International Journal of Science and Research*, 2(7), 2319–7064. Retrieved from www.ijsr.net