

Kinerja Server Basis Data pada Aplikasi Web Berbasis Raspberry Pi

Ardeman^{#1}, Sayed Muchallil^{#2}, Afdhal^{#3}

[#] *Jurusan Teknik Elektro dan Komputer, Universitas Syiah Kuala
Jl. Tgk. Syech Abdul Rauf, No. 7, Darussalam, Banda Aceh 23111 Indonesia*

¹mail@ardeman.com

²sayed.muchallil@unsyiah.ac.id

³afdhal@unsyiah.co.id

Abstrak— Raspberry Pi adalah komputer mini yang dapat berfungsi layaknya komputer PC pada umumnya. Dari hasil penelitian sebelumnya, NGINX layak untuk digunakan sebagai *web server* pada Raspberry Pi. Objek yang diuji hanya sebatas halaman statis dan dinamis saja. Permasalahannya adalah belum adanya rekomendasi *server* basis data manakah yang layak digunakan pada *platform* Raspberry Pi. Tulisan ini akan memaparkan hasil pengujian kinerja *server* basis data pada aplikasi web berbasis raspberry pi. Pengujian dilakukan terhadap tujuh aplikasi *server* basis data, yaitu: MySQL, MariaDB, Firebird, PostgreSQL, Berkeley DB, Oracle Database dan Microsoft SQL Server. Dengan menghitung waktu hasil eksekusi data di tiap perintah SQL pada PHP. Hasil pengujian menunjukkan, MySQL layak digunakan untuk penggunaan yang lebih banyak menampilkan data dari satu tabel, selain itu juga layak untuk penggunaan yang membutuhkan kecepatan proses yang stabil. Jika kebutuhannya lebih banyak menampilkan data yang menggabungkan lebih dari satu tabel disertai *input* dan *update* data, maka basis data yang layak ialah MariaDB.

Kata Kunci— Raspberry Pi, web server, NGINX, PHP, basis data.

I. PENDAHULUAN

Said F. Qiana, Mahasiswa Teknik Elektro Universitas Syiah Kuala di tahun 2014 menguji kelayakan Raspberry Pi sebagai *web server* dengan membandingkan kinerja NGINX, Apache dan Lighttpd. Penelitian tersebut menggunakan objek halaman *Hyper Text Markup Language* (HTML) statis yang berupa tampilan tulisan “Hello world...” dengan tambahan css yang membuat tulisan tersebut menjadi warna biru kemudian disertakan perintah javascript alert(“Hello world...”) ketika HTML-nya berhasil ditampilkan, dan satu halaman dinamis yang berupa perintah phpinfo() pada PHP. Ketiga *web server* yang diuji pada halaman statis berhasil melayani hingga 500 *requests* secara paralel tanpa ada kesalahan, seperti yang terlihat pada Tabel 1. Berbeda dengan hasil pengujian pada *file* dinamis, kesalahan mulai terjadi pada 200 *requests* dengan nilai kesalahan tertinggi pada *web server* Lighttpd, sedangkan Apache dapat menangani sampai dengan pada 300 *requests*. Berbeda dengan keduanya, NGINX dapat bertahan tanpa kesalahan hingga 500 *requests*. Dari hasil penelitian tersebut, NGINX

dipilih sebagai *web server* untuk mendukung penelitian ini [1].

TABLE I
PERSENTASE ERROR PADA HASIL PENGUJIAN [1]

No	Req	Apache		Lighttpd		NGINX	
		Statis	Dinamis	Statis	Dinamis	Statis	Dinamis
1	100	0	0	0	0	0	0
2	200	0	0	0	0,07	0	0
3	300	0	0,053	0	0,35	0	0
4	400	0	0,117	0	0,44	0	0
5	500	0	0,122	0	0,475	0	0

Penelitian yang dilakukan kali ini berbeda dengan penelitian yang telah dilakukan Said F. Qianna, penelitian ini justru menggunakan hasil penelitian sebelumnya untuk memilih *web server*, sedangkan aplikasi yang diuji adalah aplikasi menggunakan basis data dari perintah *Data Modified Language* (DML) berupa SELECT, INSERT, UPDATE dan DELETE dengan *record* yang berjumlah sangat besar. Namun, penelitian ini juga akan menggunakan parameter yang sama untuk mengukur kinerja aplikasi basis data atau *Relational Database Management System* (RDBMS) yang layak digunakan pada *web server* NGINX yang berjalan pada platform Raspberry Pi.

II. TINJAUAN PUSTAKA

A. Basis Data

Basis data atau *database* merupakan suatu mekanisme penyimpanan data secara terstruktur. Mekanisme tersebut memungkinkan data atau informasi yang disimpan pada basis data dapat dengan mudah untuk dipanggil kembali, ditambahkan, dimodifikasi maupun dihapus [2].

B. Relational Database Management System

Database management system (DBMS) adalah sistem yang berupa perangkat lunak, yang mengelola dan menjalankan operasi terhadap data yang diminta oleh pengguna. Sedangkan *Relational Database Management System* (RDBMS) adalah basis untuk SQL dan semua sistem basis data modern seperti MS SQL Server, IBM DB2, Oracle, MySQL dan Microsoft Access. RDBMS merupakan sistem

manajemen basis data yang berdasarkan pada model relasi yang diperkenalkan oleh E. F. Codd pada tahun 1970 [3].

C. MySQL

MySQL adalah sebuah sistem manajemen basis data (DBMS), yang berisi kumpulan data terstruktur. Isinya mulai dari daftar belanja yang sederhana hingga kedalam bentuk galeri gambar. Basis data MySQL bersifat relasional (RDBMS), yang berarti datanya dapat tersimpan secara terpisah di tabel yang berbeda tapi tetap dapat terhubung satu sama lain [2].

D. MariaDB

Secara sintaks, semua perintah yang dapat dijalankan pada MySQL juga bisa berjalan pada MariaDB. Selain itu, semua kemampuan MySQL juga dimiliki oleh MariaDB. Perbedaannya yakni, MariaDB menyertakan semua *engine open source* yang populer, tidak ada InnoDB, sebagai gantinya digunakan XtraDB. MariaDB juga mengklaim memiliki peningkatan kinerja dan kecepatan dibandingkan dengan MySQL [4].

E. Firebird

Firebird adalah sebuah RDBMS yang dikembangkan dari Inprise Corp (sekarang dikenal dengan Borland Software Corp) pada tahun 2000. Firebird sendiri telah digunakan pada sistem-sistem produksi di beberapa perusahaan sejak tahun 1981. Firebird dikembangkan dengan menggunakan bahasa C dan C++ yang dapat berjalan pada platform Linux, Windows dan bermacam jenis platform Unix [5].

F. PostgreSQL

PostgreSQL adalah salah satu dari RDBMS yang banyak digunakan selain MySQL dan Oracle. Fitur-fitur PostgreSQL yang berguna untuk replikasi basis data diantaranya DB Mirror, PGPool, Slony dan PGCluster. Selain PostgreSQL juga memiliki sistem relasi pada basis datanya, sumbernya juga bersifat terbuka (*open source*) [6].

G. Berkeley DB

Berkeley DB adalah sebuah library dari kumpulan *database embedded* yang datanya berupa *key-value*. Berkeley DB disusun menggunakan bahasa C yang dapat dipanggil menggunakan API (*Application Programming Interface*) untuk mengakses dan melakukan manajemen pada basis datanya [7].

H. Oracle Database

Oracle Database dibuat oleh Oracle Corporation, sebelumnya merupakan DBMS yang dipakai pihak militer Amerika Serikat. Namun seiring perkembangan, Oracle Database juga dipergunakan dan dijual ke pihak sipil atau bisnis [8].

I. SQL Server

SQL Server atau yang dikenal dengan Microsoft SQL Server (MS SQL) adalah RDBMS yang dikembangkan oleh

Microsoft. Produk ini berasal dari Sybase SQL Server 4.x untuk platform Unix [9].

J. Structured Query Language

Structured Query Language (SQL) adalah bahasa basis data yang dirancang untuk mengelola data dalam sistem manajemen basis data relasional. SQL awalnya dikembangkan oleh IBM pada awal tahun 1970an. SQL memiliki versi awal yang disebut SEQUEL (*Structured English Query Language*), yang mana dirancang untuk memanipulasi dan mengambil data yang disimpan dalam *IBM's quasi-relational database management system, System R*. Pada akhir tahun 1970an, Relational Software Inc., yang sekarang bernama Oracle Corporation, penggunaan SQL secara komersial pertama kali diperkenalkan dengan sebutan Oracle V2 untuk komputer VAX [10].

K. Data Definition Language

Data Definition Language atau *Data Description Language* (DDL) adalah bahasa untuk mendefinisikan sebuah data. DDL digunakan untuk membuat, mengubah struktur dan mendefinisikan tipe data dari berbagai objek basis data seperti tabel, indeks, trigger, view dan lain-lain. Contoh perintah DDL diantaranya CREATE, DROP, ALTER, RENAME dan TRUNCATE [10].

L. Data Manipulation Language

Data Manipulation Language (DML) adalah perintah yang digunakan pada basis data untuk mengolah data atau informasi. Perintah DML tersebut diantaranya INSERT, UPDATE, DELETE dan SELECT [10].

M. Data Control Language

Data Control Language (DCL) adalah perintah untuk mengatur hak akses terhadap sebuah basis data. DCL terdiri dari dua perintah yakni GRANT dan REVOKE [10].

N. Web Server

Web server adalah sebuah aplikasi *server* yang melayani *request* HTTP atau HTTPS dari peramban dan mengirimkannya kembali dalam bentuk halaman-halaman web. Halaman web yang dikirim dari *web server* biasanya berupa *file* HTML dan CSS yang nantinya akan dilakukan *parsing* atau ditata oleh peramban sehingga menjadi halaman web yang bagus dan mudah dibaca [11].

Salah satu aplikasi *web server* adalah NGINX (baca: *engine x*). NGINX dikembangkan secara *open source* oleh Igor Sysoev pada tahun 2002 dan dirilis untuk publik pada tahun 2004. NGINX tidak bergantung kepada *thread* untuk melayani *client*, karena menggunakan arsitektur asinkronus yang membutuhkan lebih sedikit *memory* sehingga NGINX dapat berjalan pada mesin VPS kecil hingga *cluster server* dalam jumlah besar [12].

O. PHP

PHP: Hypertext Preprocessor (PHP) adalah sebuah bahasa pemrograman di sisi *server*. Ketika kita mengakses sebuah URL, maka peramban web akan melakukan *request* ke sebuah *web server*. Pengembangan bahasa PHP dimulai pada tahun 1994 oleh Rasmus Lerdorf. Dia mengembangkan perkakas yang digunakan sebagai *engine parsing* sebagai penerjemah/*interpreter* beberapa *macro*. Pada saat itu engine digunakan untuk pembuatan buku tamu, *counter* dan beberapa *homepage*. Ia menamai *engine parser* tersebut dengan nama PHP/FI [13].

P. Raspberry Pi

Raspberry Pi adalah sebuah komputer berukuran kartu kredit, yang mana dapat digunakan seperti PC biasa ataupun dijadikan sebagai *server*. Jika unit Raspberry digunakan sebagai PC biasa, unit tersebut tinggal disambungkan saja dengan *monitor*, *keyboard* dan *mouse*. Dan jika dijadikan sebagai *server*, Raspberry Pi dapat langsung beroperasi dan digunakan baik di jaringan lokal (LAN) maupun jaringan internet [14].

Q. Rata-rata

Rata-rata adalah suatu nilai yang mewakili nilai suatu kelompok. Jadi rata-rata merupakan suatu hasil membagi jumlah nilai data dengan banyak data [15].

R. Standar Deviasi

Standar deviasi adalah selisih rata-rata dimana suatu nilai berada dari mean. Jadi standar deviasi merupakan ukuran tingkat pencarian selisih nilai setiap anggota dalam kelompok nilai dengan nilai rata-rata hitungannya [15].

III. METODE PENELITIAN

A. Alur Penelitian

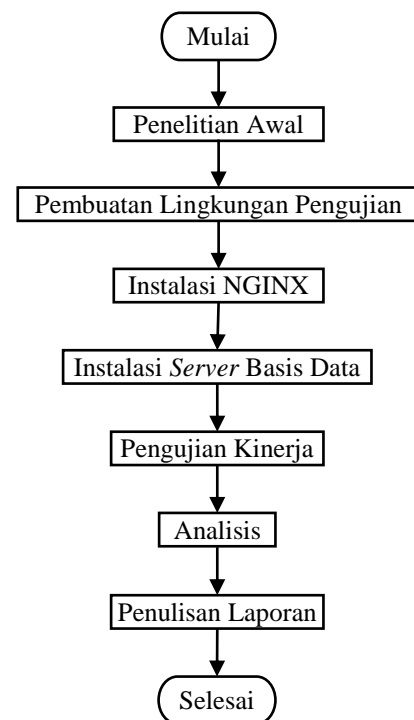
Tahapan dan metode penelitian yang dilakukan pada penelitian ini diperlihatkan pada Gambar 1.

Penelitian awal merupakan tahap pertama yang diperlukan sebagai persiapan melakukan penelitian, meliputi hal-hal seperti pengumpulan referensi baik dari jurnal, buku dan sumber lain yang berkaitan dengan penelitian ini, serta persiapan perangkat lunak dan perangkat keras yang diperlukan dalam melakukan penelitian ini.

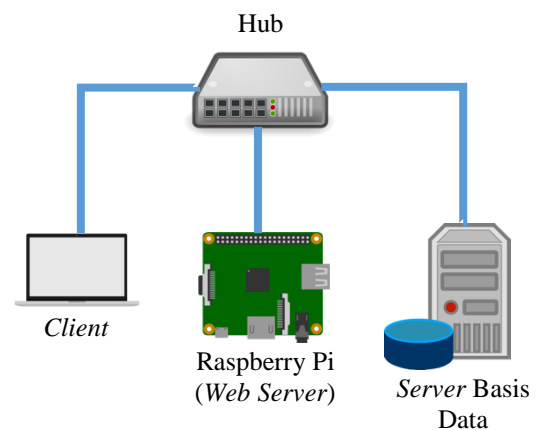
Selanjutnya dilakukan instalasi aplikasi *web server* NGINX pada Raspberry Pi dan instalasi *server* basis data MySQL, MariaDB, Firebird, PostgreSQL, Berkeley DB, Oracle Database, Microsoft SQL Server pada komputer *server* basis data dengan topologi yang diperlihatkan pada Gambar 2. Setiap *server* basis data diinstallasikan pada satu sistem operasi yang dipisahkan oleh *boot* yang berbeda (*multi boot*) agar masing-masing *server* basis data tidak mengganggu performa satu sama lain.

Setelah aplikasi *server* basis data terinstall, maka sampel basis data karyawan dieksekusi ke masing-masing *server* basis data dengan penyesuaian perintah SQLnya. Penyesuaian ini dilakukan karena setiap *server* basis data

memiliki karakteristik perintah SQL yang berbeda-beda, sedangkan sampel basis data karyawan dibuat untuk MySQL.



Gambar. 1 Blok diagram tahapan alur penelitian



Gambar 2 Topologi jaringan penelitian.

Kemudian dibuat aplikasi PHP untuk melakukan perintah DML berupa SELECT, INSERT, UPDATE dan DELETE terhadap basis data yang masing-masingnya sebanyak 300.000 baris tabel (*record*).

Jumlah aplikasi *server* basis data yang digunakan pada penelitian ini adalah sebanyak tujuh aplikasi. Untuk mengetahui kinerja dari ketujuh *server* basis data tersebut, aplikasi PHP yang telah dibuat dijalankan pada *web server* NGINX di *platform* Raspberry Pi. Pada tahap ini akan dilakukan pengujian kinerja *server* basis data menggunakan prototipe yang telah dibuat dengan 30 kali pengujian terhadap setiap perintah SELECT, INSERT, UPDATE dan DELETE pada masing-masing *server* basis data.

Pada saat melakukan pengukuran, *client* mengakses *web server* NGINX menggunakan peramban Firefox berupa halaman web yang berisi perintah PHP seperti SELECT, INSERT, UPDATE dan DELETE di tiap *filenya*. Saat NGINX diakses oleh *client*, NGINX mencatat waktu t_1 sebelum mengirimkan perintah SQL ke *server* basis data yang akan diuji. Kemudian *server* basis data akan membalas perintah SQL yang diterima ke *web server* NGINX. Balasan tersebut terlebih dahulu *diparsing* di *web server* lalu dimunculkan ke halaman web yang diminta *client* melalui *web server* NGINX. Tepat setelah *diparsing*, sebelum dikirim kembali ke *client*, waktu t_2 dicatat. Jadi, waktu eksekusi data t_2 dikurangi t_1 , pencatatan dan perhitungan waktu eksekusi data dilakukan dengan perintah PHP.

Waktu yang dihitung hanya sebatas transaksi data dari *web server* ke *server* basis data hingga datanya *diparsing* di *web server*. Waktu untuk transaksi data dari *client* hingga kembali ke *client* tidak dihitung karena akan ada parameter lain yang ikut dihitung tetapi tidak ada kaitannya dengan penelitian ini, contohnya pengaruh dari peramban yang mungkin akan berbeda waktu eksekusinya antara peramban Firefox dengan peramban Google Chrome atau peramban lainnya.

Setelah semua pengujian selesai dilakukan, langkah selanjutnya adalah pengambilan data. Data yang diambil adalah waktu eksekusi data dalam satuan detik. Data tersebut dicatat kedalam tabel data hasil eksekusi untuk setiap jenis *server* basis data yang diuji.

Pada tahap analisa data hasil yang didapatkan dari penelitian maka didapat lama waktu yang diperlukan *web server* terhadap masing-masing *server* basis data untuk menyelesaikan eksekusi SELECT, INSERT, UPDATE dan DELETE yang dilakukan. Data (total waktu eksekusi) yang diperoleh setelah melakukan pengujian terhadap MySQL, MariaDB, Firebird, PostgreSQL, Berkeley DB, Oracle Database, Microsoft SQL Server dibandingkan dengan menggunakan rata-rata dan standar deviasi.

Setelah melakukan pengujian, maka didapat data-data percobaan serta menganalisis data-data tersebut dan membandingkan antara MySQL, MariaDB, Firebird, PostgreSQL, Berkeley DB, Oracle Database, Microsoft SQL Server. Setelah hasil analisis perbandingan didapatkan, selanjutnya kesimpulan dari penelitian ini dirangkum dari segala hal yang didapatkan pada saat penelitian.

IV. HASIL DAN PEMBAHASAN

A. Kendala yang Dihadapi

Pada penelitian ini setiap *server* basis data diinstal terpisah satu sama lain agar performanya tidak saling mengganggu. Pada saat melakukan pembuatan lingkungan pengujian, ada tiga aplikasi basis data yang tidak dapat dilanjutkan untuk dilakukan pengujian karena sudah tidak sebanding kebutuhan perangkat keras yang digunakan, aplikasi basis data yang dimaksud adalah Berkeley DB, Oracle Database dan Microsoft SQL Server. Sedangkan pada saat dilakukan pengujian kinerja, ditemukan lagi basis data yang cara eksekusi perintah SQLnya yang tidak sebanding juga, yakni Firebird.

Kendala pada Berkeley DB adalah aplikasi ini yang tidak dapat berdiri sendiri. Maksudnya, Berkeley DB harus dijalankan pada mesin yang sama dengan aplikasi yang menggunakannya. Ini akan mempengaruhi kinerja *web server* NGINX yang berjalan pada platform Raspberry Pi. Hal ini diperkuat dengan perintah koneksi PHP dengan Berkeley DB yang memerlukan akses langsung ke *file* database langsung dari perintah PHPnya.

Pada Oracle Database, kendalanya adalah *library* yang disediakan oleh Oracle (Oracle Instant Client) belum mendukung perangkat Raspberry Pi dengan prosesor ARM [16].

Sedangkan pada Microsoft SQL Server, spesifikasi komputer yang dijadikan *server* basis data belum memenuhi persyaratan yang dibutuhkan oleh SQL Server. Persyaratan yang dibutuhkan yang tidak cocok dengan spesifikasi komputernya adalah [17]:

- Sistem operasi ubuntu yang didukung versi 16.04 dan 16.10, sedangkan versi yang digunakan pada penelitian ini adalah 14.04.1.
- Minimum RAM 3,25 GB, sedangkan RAM yang digunakan pada penelitian ini sebesar 1 GB.

Pada Firebird, INSERT tidak dapat dilakukan secara *multirow*, yakni dengan cara melakukan input data lebih dari satu baris dengan sekali perintah INSERT. Alternatif lainnya, dilakukan dengan cara menggabungkan perintah INSERT dengan satu BLOCK transaksi, tapi hanya dapat dilakukan dengan 255 baris sekaligus. Akhirnya dengan cara ini, penelitian dilanjutkan dengan melakukan perulangan tiap 255 baris untuk 300.000 baris tabel. Dengan berbedanya perlakuan INSERT pada Firebird, mengakibatkan rendahnya performa yang dihasilkan oleh Firebird.

B. Rata-rata

Berdasarkan hasil Tabel 2 rata-rata waktu eksekusi data, pada pengujian terhadap SELECT satu tabel, performansi MySQL lebih cepat. Pada pengujian SELECT dua tabel, SELECT tiga tabel, INSERT dan UPDATE, performansi MariaDB lebih cepat. Sedangkan pada pengujian terhadap DELETE, performansi PostgreSQL yang lebih cepat. Khusus untuk INSERT, performansi Firebird menjadi sangat lambat dikarenakan proses INSERT yang dilakukan menggunakan perulangan setiap 255 baris data.

TABLE II
RATA-RATA WAKTU EKSEKUSI DATA

Pengujian	Rata-rata Waktu Eksekusi Data (detik)			
	MySQL	MariaDB	Firebird	PostgreSQL
SELECT satu tabel	52,187	55,632	124,038	81,545
SELECT dua tabel	75,109	73,403	148,659	122,142
SELECT tiga tabel	89,803	75,99	166,472	130,332
INSERT	24,522	23,949	413,724	123,416
UPDATE	13,342	11,818	71,576	18,6
DELETE	6,883	5,901	86,253	3,146

C. Standar Deviasi

Berdasarkan Tabel 3, MySQL ketika pengujian SELECT satu tabel memiliki standar deviasi sebesar 3,542. Pada pengujian SELECT dua tabel, besar nilai standar deviasi yaitu 2,895 dan pada SELECT tiga tabel nilainya 3,826. Ketika pengujian INSERT besar standar deviasinya yaitu 0,661, UPDATE sebesar 0,58 dan DELETE sebesar 0,512. Berdasarkan nilai standar deviasi MySQL dari SELECT satu tabel hingga DELETE, tampak nilai standar deviasi berkisar antara 0,51 hingga 3,89. Jadi nilai standar deviasi pada MySQL memiliki selisih standar deviasi yang kecil, dan nilai standar deviasi yang stabil yaitu nilai standar deviasi yang bernilai nol atau mendekati nol.

TABLE III
RATA-RATA WAKTU EKSEKUSI DATA

Pengujian	Rata-rata Waktu Eksekusi Data (detik)			
	MySQL	MariaDB	Firebird	PostgreSQL
SELECT satu tabel	52,187	55,632	124,038	81,545
SELECT dua tabel	75,109	73,403	148,659	122,142
SELECT tiga tabel	89,803	75,99	166,472	130,332
INSERT	24,522	23,949	413,724	123,416
UPDATE	13,342	11,818	71,576	18,6
DELETE	6,883	5,901	86,253	3,146

Pada MariaDB, ketika pengujian SELECT satu tabel besar nilai standar deviasi yaitu 2,272. Ketika SELECT dua tabel nilai standar deviasi yaitu 4,165. Ketika SELECT tiga tabel besar nilai standar deviasi yaitu 3,251 dan ketika pengujian INSERT, UPDATE dan DELETE nilai standar deviasi masing-masingnya yaitu 1,892, 0,629 dan 0,644. Berdasarkan nilai standar deviasi MariaDB dari SELECT satu tabel hingga DELETE tampak nilai standar deviasi berkisar antara 0,63 hingga 4,16. Jadi nilai standar deviasi pada MariaDB memiliki selisih standar deviasi yang kecil,

dan nilai standar deviasi yang stabil yaitu nilai standar deviasi yang bernilai nol atau mendekati nol.

Ketika pengujian SELECT satu tabel pada Firebird, besar nilai standar deviasi yaitu 1,585. Pada SELECT dua tabel nilai standar deviasi yaitu 1,092. Ketika SELECT tiga tabel, nilai standar deviasinya 1,558. Khusus untuk INSERT, standar deviasinya melonjak ke angka 108,77. Untuk pengujian UPDATE standar deviasinya turun ke 7,494 dan pada pengujian DELETE nilai standar deviasinya 7,415. Berdasarkan nilai standar deviasi Firebird dari SELECT satu tabel hingga DELETE tampak nilai standar deviasi berkisar antara 1,09 hingga 108,77. Jadi nilai standar deviasi pada Firebird memiliki selisih standar deviasi yang sangat besar. Namun, jika pengujian INSERT diabaikan karena perbedaan perlakuan dengan pengujian INSERT pada basis data lainnya, maka standar deviasinya berkisar antara 1.09 hingga 7,49.

Pada PostgreSQL ketika pengujian SELECT satu tabel, nilai standar deviasinya sebesar 1,883. Ketika pengujian SELECT dua tabel, nilai standar deviasinya sebesar 11,255. Ketika pengujian SELECT tiga tabel, nilai standar deviasinya sebesar 6,494. Ketika pengujian INSERT besar nilai standar deviasinya yaitu 41,048. Ketika pengujian UPDATE besar nilai standar deviasinya yaitu 5,539 dan pada pengujian DELETE besar nilai standar deviasinya yaitu 4,06. Berdasarkan nilai standar deviasi PostgreSQL dari SELECT satu tabel hingga DELETE tampak nilai standar deviasi berkisar antara 1,88 hingga 41,05. Jadi nilai standar deviasi pada PostgreSQL memiliki selisih standar deviasi yang cukup besar.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan: Pertama, Perbandingan kinerja server basis data pada aplikasi web berbasis Raspberry Pi telah selesai dilaksanakan menggunakan 1 *web server*, 1 *server* basis data dan 1 *client* dengan parameter yang dibandingkan adalah total waktu eksekusi.

Kedua, Aplikasi *server* basis data yang dilakukan pengujian ini adalah MySQL, MariaDb, Firebird, PostgreSQL, Berkeley DB, Oracle Database dan Microsoft SQL Server. Sedangkan aplikasi *web server* yang digunakan hanya NGINX atas hasil dari penelitian sebelumnya yang dilakukan oleh Said Fairuz Qiana, Mahasiswa Teknik Elektro Universitas Syiah Kuala pada tahun 2014.

Ketiga, Pada saat melakukan pembuatan lingkungan pengujian, Berkeley DB, Oracle Database dan Microsoft SQL Server tidak dapat dilanjutkan penelitiannya karena kebutuhan perangkat keras yang tidak sebanding. Sedangkan pada saat melakukan pengujian kinerja, cara eksekusi perintah INSERT pada Firebird tidak sebanding perintah SQLnya.

Keempat, Hasil percobaan pada saat uji SELECT satu tabel, dapat disimpulkan bahwa MySQL lebih cepat dibandingkan dengan yang lain. Pada pengujian SELECT dua tabel, SELECT tiga tabel, INSERT dan UPDATE, performa MariaDB yang lebih cepat. Sedangkan pada

pengujian DELETE, PostgreSQL lebih cepat daripada yang lain.

Kelima, Untuk kestabilan performa, MySQL lebih stabil dengan selisih standar deviasi antara 0,51 hingga 3,89.

Terakhir, saran untuk pengujian berikutnya, ketika melakukan pengujian terhadap MySQL, MariaDB, Firebird, PostgreSQL, Berkeley DB, Oracle Database dan Microsoft SQL Server dengan spesifikasi RAM 1 GB dan sistem

operasi Ubuntu Server 14.04.1, tidak semua aplikasi *server* basis data berhasil dilakukan instalasinya. Oleh karena itu perlu spesifikasi dan sistem operasi yang sebanding dan semuanya mencukupi untuk dilakukan instalasi dengan baik sehingga jika ingin melakukan penelitian berikutnya dengan menggunakan aplikasi basis data yang sama tidak mengalami kendala yang sama.

REFERENSI

- [1] R. Dawood, S. F. Qiana, and S. Muchallil, "Kelayakan Raspberry Pi sebagai web server: Perbandingan kinerja Nginx, Apache, dan Lighttpd pada platform Raspberry Pi," *J. Rekayasa Elektr.*, vol. 11, no. 1, pp. 25–29, 2014.
- [2] A. Solichin, *MySQL Dari Pemula Hingga Mahir*. Jakarta: Universitas Budi Luhur, 2010.
- [3] S. Sumathi and S. Esakkirajan, *Fundamentals of Relational Database Management Systems*, vol. 47. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.
- [4] D. Bartholomew, *Getting Started with MariaDB*. Birmingham: Packt Publishing Ltd, 2015.
- [5] P. Reeves, "Whats Happening to InterBase?," 2000. [Online]. Available: http://www.ibphoenix.com/resources/documents/search/doc_14. [Accessed: 15-Mar-2017].
- [6] L. A. Rowe and M. R. Stonebraker, "The POSTGRES Data Model," in *Proceedings of the 13th International Conference on Very Large Data Bases*, 1987, pp. 83–96.
- [7] M. A. Olson, K. Bostic, and M. Seltzer, "Berkeley DB," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, 1999, pp. 43–43.
- [8] H. Kusuma, *Database Oracle Untuk Pemula*, 1st ed. Jakarta: Eastern Light Publication, 2011.
- [9] M. C. Amri, *Pengantar Administrasi Microsoft SQL Server 2000*. IlmuKomputer.Com, 2003.
- [10] J. A. Watt and N. Eng, *Database Design*, 2nd ed. British Columbia: B.C. Open Textbook Project, 2014.
- [11] S. Silva, *Web Server Administration*. Massachusetts: Course Technology, 2011.
- [12] D. Sarkar, *Nginx 1 Web Server Implementation Cookbook*. Birmingham: Packt Publishing Ltd, 2011.
- [13] I. Daqiqil, *Framework Codeigniter*. Pekanbaru: Koder, 2011.
- [14] Fadhil, "Analisis Kinerja Cluster Nginx Berbasis Raspberry Pi," Universitas Syiah Kuala, 2014.
- [15] J. Stanbrough, "Standard Deviation," *Batesville*, 2004. [Online]. Available: http://www.batesville.k12.in.us/physics/apphynet/Measurement/standard_deviation.htm. [Accessed: 05-Jun-2017].
- [16] "Oracle Instant Client," *Oracle*. [Online]. Available: <http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>. [Accessed: 05-Jun-2017].
- [17] J. Roth, C. Mansfield, M. Ray, and L. Bosquez, "Install SQL Server on Linux," *Microsoft*, 2017. [Online]. Available: <https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-setup#system>. [Accessed: 05-Jun-2017].