

## STUDI KOMPARASI MENYIMPAN DAN MENAMPILKAN DATA HISTORI ANTARA DATABASE TERSTRUKTUR MARIADB DAN DATABASE TIDAK TERSTRUKTUR INFLUXDB

Hendra<sup>1</sup>, Widyastuti Andriyani<sup>2</sup>

<sup>1</sup>Mahasiswa Magister Teknologi Informasi, STMIK AKAKOM Yogyakarta

<sup>2</sup>Magister Teknologi Informasi, STMIK AKAKOM Yogyakarta

Jl. Raya Janti (Majapahit) No. 143, Karangjambe, Yogyakarta

E-mail <sup>1</sup>camp.hendra@gmail.com, <sup>2</sup>widya@akakom.ac.id

Masuk: 31 Januari 2020, Revisi masuk: 18 Februari 2020, Diterima: 19 Februari 2020

### ABSTRACT

*The use of structured databases is still very widely used by companies in small and medium scale with the aim of processing data so that from these data conclusions can be drawn to determine a decision. But over time, of course the need for data that continues to grow can make a system run very slowly when using a structured database. That is caused by the amount of data that continues to increase every day, even for certain cases the data can increase every second. For this reason, an unstructured database is needed specifically for storing history data. From some existing unstructured databases, InfluxDB is one of the unstructured databases specifically intended for storing history data and has a very good ability to process data into a matrix for analysis. One of the key factors in an unstructured database is the database structure which is very different and supports to maximize database performance.*

**Keywords:** *Big data, InfluxDB, MariaDB, Time-series data.*

### INTISARI

Penggunaan *database* terstruktur masih banyak digunakan oleh perusahaan dalam skala kecil dan menengah dengan tujuan untuk melakukan pemrosesan data sehingga dari data tersebut dapat diambil kesimpulan untuk menentukan sebuah keputusan. Namun seiring berjalannya waktu, kebutuhan data yang terus menerus bertambah dapat membuat sebuah sistem menjadi berjalan lambat jika menggunakan *database* terstruktur. Hal itu disebabkan oleh jumlah data yang terus menerus bertambah setiap hari, bahkan untuk kasus tertentu data dapat bertambah setiap detik. Untuk itu diperlukan sebuah *database* tidak terstruktur yang khusus digunakan untuk menyimpan data histori. Dari beberapa *database* tidak terstruktur yang ada, InfluxDB merupakan salah satu *database* tidak terstruktur yang khusus diperuntukkan untuk menyimpan data histori dan memiliki kemampuan yang baik dalam memproses data ke dalam bentuk matriks untuk dianalisis. Salah satu faktor kunci dalam *database* tidak terstruktur adalah struktur *database* yang sangat berbeda dan mendukung untuk memaksimalkan performa *database*.

**Kata-kata Kunci:** *Big Data, InfluxDB, MariaDB, Time-Series Data.*

### PENDAHULUAN

Perkembangan dunia teknologi saat ini sangat cepat, sehingga manusia dituntut untuk mengikuti perkembangan. Untuk itu dibutuhkan sumber daya manusia dalam bidang IT yang memiliki kemampuan beradaptasi secara cepat. Dalam dunia teknologi informasi saat ini kebutuhan terhadap data adalah sebuah kunci yang wajib dimiliki pada setiap lini bisnis. Meskipun bisnisnya bahkan tidak berhubungan dengan dunia teknologi, namun data sangat diperlukan dalam melakukan analisis yang nantinya akan digunakan untuk pengambilan keputusan.

Data histori atau biasa disebut juga dengan *time series data* merupakan sebuah data yang disimpan ke dalam *database* dengan tujuan untuk mencatat sejarah pergerakan angka [1]. Biasanya jenis data seperti ini tidak membutuhkan jumlah kolom yang banyak, karena hanya mencatat nama variabel, nilai variabel, dan waktu. Namun karena tujuan dari data histori adalah mencatat pergerakan angka yang terjadi setiap menit bahkan setiap detik, tentunya pemilihan *database* merupakan bagian yang sangat penting, karena *database* tersebut nantinya harus mampu mengelola data dalam jumlah yang sangat besar sehingga performa *database* tetap terjaga.

Salah satu *database* yang cukup populer hingga saat ini adalah MariaDB dimana *database* ini adalah *database* yang dibuat oleh orang yang sama dengan yang membuat MySQL sehingga semua struktur *database* MariaDB sangat mirip dengan MySQL [2]. Salah satu alat bantu pendukung yang digunakan pada MariaDB adalah PHPMyAdmin yang berfungsi untuk melakukan semua administrasi terhadap *database* dan *user* secara visual yang berjalan menggunakan *platform* berupa *browser* [3].

Saat ini tercatat sudah cukup banyak perusahaan besar yang bermigrasi ke *database* MariaDB, contohnya adalah perusahaan komunikasi Verizon, Samsung, DBS Bank, Deutsche Bank, Nasdaq, ingenico Payment Service, dan masih banyak lagi.

InfluxDB merupakan sebuah *database* khusus yang dibuat untuk menyimpan data *time series* [4]. Data *time series* merupakan sebuah data yang disimpan lengkap dengan data waktu (*timestamp*) untuk menunjukkan kapan data tersebut disimpan.

Tujuan dari penelitian ini adalah untuk memberikan analisis perbandingan kinerja dalam menyimpan dan menampilkan data antara MariaDB dengan InfluxDB.

Penelitian tentang perbandingan kinerja *database* juga telah dilakukan dengan membandingkan kinerja *query* pada *database management system* (DBMS) antara MySQL 5.7.16 dan MariaDB 10.1 [5]. Dalam penelitian tersebut dapat dilakukan perbandingan kinerja *database* secara *apple to apple* karena secara struktur kedua *database* tersebut adalah sama.

Selain itu juga terdapat penelitian yang serupa yang membandingkan kinerja antara NoSQL yang diwakili dengan MongoDB dan SQL yang diwakili oleh MySQL [6]. Hasil dari penelitian ini menunjukkan bahwa performa MongoDB lebih baik dibandingkan dengan MySQL pada saat menangani jumlah baris data yang cukup besar yaitu 226.232 *record*.

Penelitian lain yang memberikan informasi secara detail adalah sebuah paper yang ditulis oleh Kumar dan Jayagopal [7]. Dalam paper tersebut benar-benar dibandingkan secara utuh antara *database* NoSQL dengan *database* tradisional. Hasil dari penelitian tersebut menyebutkan bahwa NoSQL memiliki keuntungan yang besar dari segi performa menangani data dalam jumlah yang besar dibandingkan dengan *database* tradisional.

## METODE PENELITIAN

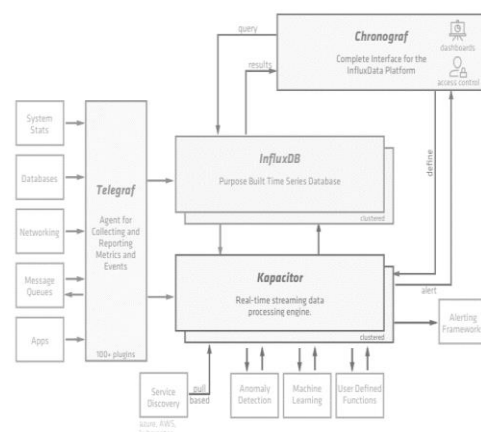
Metode yang digunakan dalam penelitian ini adalah metode komparatif dimana dilakukan perbandingan berdasarkan spesifikasi, cara kerja dan fitur kunci dari masing-masing *database*.

Perbandingan secara *head-to-head* dilakukan untuk menemukan persamaan dan perbedaan dengan melakukan pengamatan langsung terhadap objek masing-masing *database* dan juga berdasarkan dokumentasi resmi yang di *publish* di website masing-masing *database*.

## HASIL DAN PEMBAHASAN InfluxDB

InfluxDB adalah salah satu *time-series database* yang cukup populer saat ini, berbasis *open source* yang dikembangkan oleh InfluxData dengan bahasa pemrograman Go. InfluxDB memiliki kecepatan yang cukup tinggi dalam melakukan *read data* maupun *write data* sehingga cocok jika diterapkan pada aplikasi yang bertipe *real-time* [8], baik itu untuk penerapan model *machine learning* maupun melakukan penyimpanan data dari IoT [9]. InfluxDB dibuat untuk menangani penulisan data dan *query* muatan yang masif atau tinggi dengan menggunakan HTTP *Representational State Transfer* (REST) API untuk proses *query* dan menggunakan *SQL-like query language*.

Secara garis besar InfluxDB dibagi ke 3 komponen utama yaitu Grafana, InfluxDB, dan Telegraf. Skema model InfluxDB ditunjukkan pada Gambar 1.



Gambar 1. Skema model InfluxDB

Telegraf merupakan sebuah *plugin* atau bisa disebut juga *engine* yang berfungsi sebagai pengambil data baik secara *streaming* (*IoT sensor*, *log data*, *real-time analytics*, *metriks*, dsb) maupun mengambil

data yang bersumber dari data statis (*database* seperti *MongoDB*, *MySQL*, *Cassandra*, *PostgreSQL*, *Apache HBASE*, dsb). Telegraf dapat mengumpulkan metrik dari beragam input dan menuliskannya ke dalam beragam output. Telegraf merupakan *plugin-driven* untuk pengumpulan dan keluaran data sehingga mudah diterapkan dalam berbagai konsep maupun model.

Telegraf ditulis dengan bahasa pemrograman *Go* yang berarti bahwa ini dapat dieksekusi pada sistem apapun tanpa memerlukan ketergantungan eksternal, tidak memerlukan *npm*, *pip*, *gem*, atau alat manajemen paket lainnya yang diperlukan. Data yang disimpan di dalam *InfluxDB* bersifat *key-value*, dimana *key* berupa waktu (*timestamps*) dengan format *nanosecond* dan *value* berupa data yang pada umumnya bertipe numerik. Setiap data akan disimpan ke dalam *measurement* (dalam *database* tradisional disebut juga tabel) akan diberikan *tags* yang akan membantu mengelompokkan data berdasarkan *tags* tertentu (disebut juga kategori).

Komponen selanjutnya yang menjadi unggulan dari *InfluxDB* adalah *Chronograf*, dimana fungsi utamanya adalah sebagai *user interface* dan *administrative component* pada platform *InfluxDB*. *Chronograf* memungkinkan penggunaannya untuk melihat data yang disimpan di dalam *database* *InfluxDB* dan membangun sebuah *query* atau bahkan menciptakan peringatan. *Chronograf* cukup mudah digunakan dan dilengkapi *templates* dan *library* sehingga mudah untuk membuat visualisasi data secara *real-time*.

*InfluxDB* secara garis besar juga disebut sebagai *Push system*, dimana *InfluxDB* akan melakukan transformasi data ke arah visualisasi untuk keperluan *monitoring* data.

### MariaDB

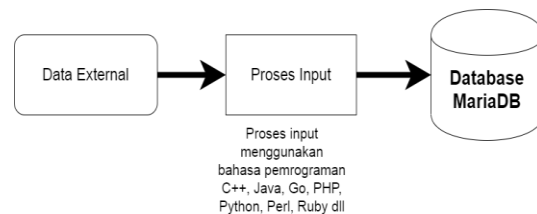
*MariaDB* merupakan *Relational Database Management System* (RDBMS) yang cukup populer dikarenakan diciptakan oleh orang yang sama dengan yang menciptakan *MySQL*. Secara umum baik perintah, fungsi, maupun tampilan sangat mirip antara *MySQL* dengan *MariaDB*.

*MariaDB* memiliki struktur *database* berupa relasi antara tabel yang satu dengan yang lain. Dalam artian bahwa sebelum menggunakan *MariaDB*, terlebih dahulu harus dilakukan inisialisasi terhadap setiap tabel dan setiap *field* di dalam masing-masing tabel. Setelah dilakukan inisialisasi, baru kemudian dilakukan *normalisasi* tabel untuk memaksimalkan performa *database*.

*Normalisasi* tabel di dalam *database* *MariaDB* adalah sebuah proses yang dilakukan untuk mengelompokkan atribut data sehingga nantinya data yang akan dihasilkan tidak terjadi *redundansi* data dan memastikan data berada pada tabel yang benar. Proses ini sekaligus mencegah terhadap anomali data yang kemungkinan terjadi di kemudian hari yang tentunya merugikan sistem secara keseluruhan, khususnya terkait dengan kinerja *database*.

### Perbandingan MariaDB dan InfluxDB

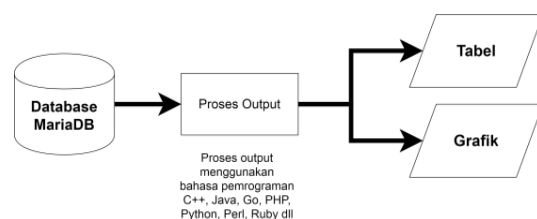
Karena terdapat perbedaan yang sangat mendasar antara *InfluxDB* dan *MariaDB*, maka untuk membedakannya akan dibagi ke dua bagian, yaitu perbedaan dalam menyimpan data dan dalam menampilkan data. Gambar 2 menampilkan skema input data pada *MariaDB*.



Gambar 2. Skema input data *MariaDB*

*MariaDB* memiliki *tools* yang berfungsi untuk visualisasi *database* bernama *phpmyadmin*. Dengan *tools* ini data dapat dimasukkan langsung ke alam tabel melalui menu *insert*. Namun pada prakteknya *tools* *phpmyadmin* jarang digunakan, karena untuk proses *input* data ke *database* *MariaDB* dibutuhkan *custom-program* yang terhubung dengan *database* *MariaDB*. Beberapa bahasa pemrograman yang bisa bekerja dengan *MariaDB* antara lain *C++*, *Perl*, *Java*, *PHP*, dan *Python*. Program yang telah dibuat tersebut yang nantinya akan digunakan untuk proses input data ke dalam *database*.

Sementara untuk menampilkan data juga dibutuhkan *custom-program* yang berfungsi untuk *load data* dari dalam *database* yang kemudian akan divisualisasikan dalam bentuk tabel ataupun grafik. Gambar 3 menampilkan skema output data pada *MariaDB*.



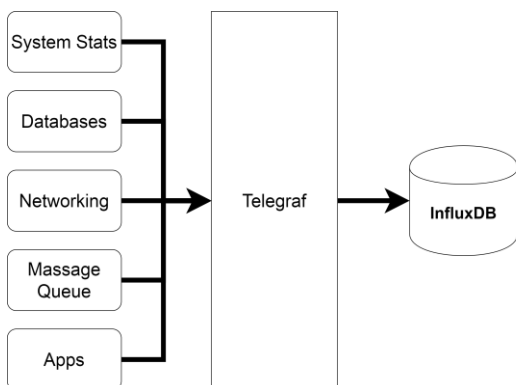
Gambar 3. Skema output data *MariaDB*

MariaDB hanya berfokus terhadap penyimpanan data, sementara untuk proses input data dan menampilkan data membutuhkan bantuan *custom-program* dengan berbagai bahasa pemrograman yang cukup banyak dan memberikan pilihan sesuai dengan kebutuhan sistem.

InfluxDB memberikan pengalaman yang berbeda, dimana setelah menciptakan sebuah *database*, data sudah bisa langsung dimasukkan tanpa perlu dilakukan konfigurasi terlebih dahulu. Untuk input data InfluxDB juga menyediakan *query* yang bahasanya sangat mirip dengan SQL, sehingga *database* ini bisa jika menggunakan *custom-program*.

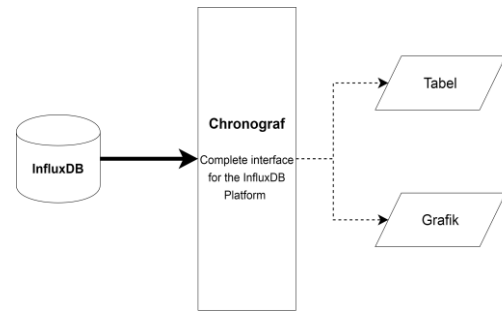
Namun, jika data diambil secara *streaming*, baik itu dari IoT, *website*, *message queue*, atau *networking*, maka bisa digunakan *telegraf*. Dengan adanya fitur ini, pengguna cukup mengunduh *plugin telegraf*, dan melakukan konfigurasi didalamnya, maka *telegraf* akan melakukan pengambilan data secara *streaming* dan langsung memasukkan data ke dalam *database*.

*Telegraf* akan *online* secara terus menerus melakukan pengambilan data serta langsung melakukan input data ke dalam *database* selama tidak dimatikan. Skema ini akan sangat cocok terhadap data IoT lalu lintas, cuaca, pergerakan harga saham, jaringan komputer dan lain sebagainya. Gambar 4 menampilkan skema input data InfluxDB.



Gambar 4. Skema input data InfluxDB

Setelah data dimasukkan ke dalam *database* InfluxDB, yang menjadi fokus utama selanjutnya adalah menampilkan data ke dalam bentuk yang mudah dipahami. InfluxDB memiliki *tools* yang disebut dengan *chronograf* yang berfungsi memvisualisasikan data ke bentuk tabel ataupun grafik [10]. Gambar 5 menampilkan skema output data InfluxDB.



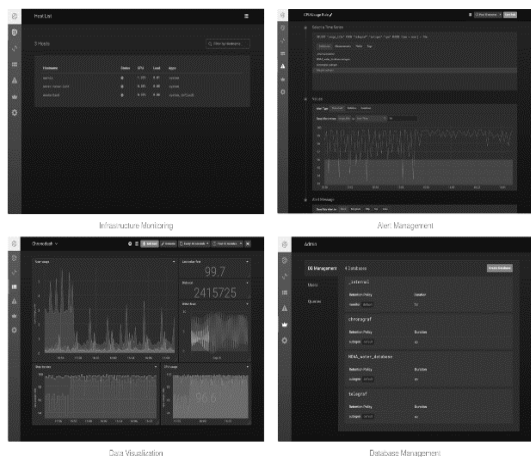
Gambar 5. Skema output data InfluxDB

*Chronograf* mudah untuk digunakan karena pengguna cukup melakukan konfigurasi terhadap data apa saja yang ingin ditampilkan, kemudian visualisasi dalam bentuk tabel atau grafik, bahkan *chronograf* memiliki fitur untuk memberikan *warning* jika ada data tertentu yang melebihi batas kewajaran. Fitur ini sangat membantu untuk pengawasan data, karena dalam keadaan yang sesungguhnya seorang pengguna yang bertugas mengawasi tidak mungkin akan selalu *stand-by* di depan monitor.

Selain menggunakan *chronograf*, data pada InfluxDB bisa di-load ke *custom-program* dengan menggunakan berbagai bahasa pemrograman seperti Go, Java, PHP, R, Scala, dan lain sebagainya. Menampilkan data menggunakan *database* InfluxDB memiliki fleksibilitas yang tinggi, baik menggunakan aplikasi bawaan dari InfluxDB yaitu *chronograf*, maupun menggunakan bahasa pemrograman yang sudah ada sebelumnya.

*Chronograf* memiliki 3 fitur utama. yaitu visualisasi data, peringatan terhadap data, dan monitor terhadap infrastruktur. Ketiga fitur ini akan sangat membantu dalam pembuatan sebuah sistem, karena untuk mendapatkan ketiga fitur tersebut tidak perlu dilakukan *coding* program lagi.

Gambar 6 menampilkan cuplikan tampilan *chronograf* yang menampilkan data secara tabel ataupun grafik. Selain grafik yang bersifat statis, *chronograf* juga bisa menampilkan *running* grafik yang bisa di-*update* sesuai keinginan (misalkan per detik atau setiap ada data baru).



Gambar 6. Tampilan Chronograf

Tabel 1 menampilkan perbandingan antara *database* MariaDB dan InfluxDB yang memberikan gambaran secara lebih detail dari kedua *database*. Sedangkan Tabel 2 menampilkan perbandingan perintah dasar pada InfluxDB dan MariaDB.

Tabel 1. Perbandingan *head-to-head* InfluxDB dan MariaDB

	MariaDB	InfluxDB
Deskripsi	RDBMS “ <i>Open Source</i> ” dengan tingkat keamanan dan kinerja yang cukup tinggi. Bekerja maksimal dengan penyimpanan data berorientasi kolom.	DBMS yang khusus digunakan untuk menyimpan data “ <i>Time Series</i> ”, “ <i>Events</i> ” dan “ <i>Metrics</i> ”.
Model Database Utama	Relasional DBMS	<i>Time Series DBMS</i>
Website	<a href="http://mariadb.com">mariadb.com</a> & <a href="http://mariadb.org">mariadb.org</a>	<a href="http://www.influxdata.com/products/influxdb-overview">www.influxdata.com/products/influxdb-overview</a>
Website Dokumentasi	<a href="http://mariadb.com/kb/en/library">mariadb.com/kb/en/library</a>	<a href="http://docs.influxdata.com/influxdb">docs.influxdata.com/influxdb</a>
Peluncuran Pertama	2009	2013
Lisensi	<i>Open Source</i>	<i>Open Source</i>
Bahasa	C & C++	Go
Tersedia di Server	<i>FreeBSD</i> <i>Linux</i> <i>Solaris</i> <i>Windows</i>	<i>Linux</i> <i>OS X</i>
Skema Data	Ya	<i>Skema Bebas</i>
Definisi Data di awal	Ya	<i>Numeric data and Strings</i>
Dukungan XML	Ya	Tidak
Dukungan Bahasa SQL	Ya	SQL seperti bahasa Query
Dukungan API & Metode akses data	<i>ADO.NET</i> <i>JDBC</i> <i>ODBC</i> <i>Proprietary native API</i>	<i>HTTP API</i> <i>JSON over UDP</i>
Dukungan Bahasa Pemrograman	<i>Ada</i> <i>C</i> <i>C#</i> <i>C++</i> <i>D</i> <i>Eiffel</i> <i>Erlang</i> <i>Go</i> <i>Haskell</i> <i>Java</i> <i>JavaScript (Node.js)</i> <i>Objective-C</i>	<i>.Net</i> <i>Clojure</i> <i>Erlang</i> <i>Go</i> <i>Haskell</i> <i>Java</i> <i>JavaScript</i> <i>JavaScript (Node.js)</i> <i>Lisp</i> <i>Perl</i> <i>PHP</i> <i>Python</i>

	MariaDB	InfluxDB
	Ocaml Perl PHP Python Ruby Scheme Tcl	R Ruby Rust Scala
Metode Partisi	Horizontal partitioning, sharding with Spider storage engine or Galera cluster	Sharding
Metode Replikasi	Master-master replication Master-slave replication	Selectable replication factor

Tabel 2. Perbandingan perintah dasar antara InfluxDB dan MariaDB

	MariaDB	InfluxDB
Menciptakan Database Baru	CREATE DATABASE IF NOT EXISTS `[nama_database]` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;	CREATE DATABASE [nama_database]
Menciptakan Tabel Baru	CREATE TABLE [nama_tabel] ([Nama_Field1] [type] NOT NULL, [Nama_Field2] [type] NOT NULL, ...)	None
Menyimpan Data	INSERT INTO [nama_tabel] (kolom1, kolom2, kolom3, ...) VALUES ('data1', 'data2', 'data3', ...)	INSERT [nama_measurement], nama_tag_set=[tag_set], nama_value=[value] [timestamp]
Mengupdate Data	UPDATE [nama_tabel] SET kolom1=[data_baru], kolom2=[data_baru2], ... WHERE [kondisi]	Simpan data baru dengan [timestamp] yang sama terhadap data lama yang ingin diperbaharui.
Menghapus Data	DELETE FROM [nama_tabel] WHERE [kondisi]	DROP SERIES FROM [nama_measurement] WHERE [tag_key] = [tag_value]
Menampilkan Data	SELECT * FROM [nama_tabel]	SELECT * FROM [measurement]

## KESIMPULAN

Setiap *database* memiliki kelebihan dan kekurangannya masing-masing dalam mengelola, memproses, dan menampilkan data. Secara umum *database* MariaDB dan InfluxDB tidak dapat dibandingkan secara *head-to-head* karena infrastruktur kedua *database* tersebut dibuat dengan sangat berbeda jauh.

InfluxDB memiliki keunggulan dalam menyimpan data *time-series* dikarenakan beberapa hal sebagai berikut:

1. InfluxDB mendukung format *time* hingga ke *nanosecond*, sementara MariaDB hanya mendukung format *time* hingga *second*.
2. InfluxDB tidak perlu melakukan inialisasi data di awal, sementara MariaDB perlu melakukan inialisasi data terlebih dahulu sebelum digunakan. Hal ini menunjukkan bahwa MariaDB tidak siap terhadap

perubahan data di kemudian hari seperti penambahan *field*, karena MariaDB harus menginisialisasi *field* baru terlebih dahulu. Sementara InfluxDB cukup menambahkan *tags* baru pada proses input data tanpa perlu inialisasi terlebih dahulu.

3. InfluxDB sudah dilengkapi *tools* yang bisa melakukan input data secara *streaming* serta menampilkan data dengan visualisasi yang bisa disesuaikan dengan keinginan, sementara MariaDB adalah *database* yang berdiri sendiri, sehingga dibutuhkan program *custom* tambahan untuk menyimpan data, memproses data, dan memvisualisasikan data ke bentuk yang lebih mudah dipahami.
4. InfluxDB lebih fleksibel untuk melakukan input maupun output data dalam bentuk visualisasi, karena selain bisa menggunakan *tools* bawaan

seperti *telegraf* dan *chronograf* juga dapat dibuat program dengan berbagai bahasa pemrograman yang tersedia, sementara MariaDB dalam melakukan input data dan menampilkan data tidak menyediakan *tools* bawaan, melainkan harus dibuat program sendiri yang didukung oleh bahasa pemrograman yang banyak tersedia dan populer saat ini.

Selain itu juga dapat disimpulkan bahwa InfluxDB hanya bisa maksimal dalam penyimpanan data bertipe *time-series* yang mendukung format *key-value*, sehingga jika *database* InfluxDB digunakan untuk kebutuhan lain maka hasilnya tidak akan maksimal (misalkan untuk *database* administrasi perusahaan), sementara MariaDB memiliki kemampuan yang sangat baik jika digunakan untuk kebutuhan selain *time-series*.

Sebagai lanjutan penelitian ini, diharapkan dapat menampilkan perbandingan performa kedua *database* tersebut dalam menangani data *time-series* baik dalam penyimpanan maupun menampilkan data. Dengan adanya perbandingan performa ini, maka akan dapat memberikan *insight* seberapa kuat sebuah *database* dari sisi kecepatan jika menangani data dalam jumlah besar.

#### DAFTAR PUSTAKA

- [1] P. Esling and C. Agon, "Time-series Data Mining," ACM Comput. Surv., 2012.
- [2] MariaDB Foundation, "About Mariadb," mariadb.org, 2016.
- [3] "phpMyAdmin," in The Definitive Guide to MySQL5, 2006.
- [4] Dix, P., "InfluxData (InfluxDB) Time Series Database Monitoring & Analytics," InfluxData, Inc., 2017.
- [5] I. Warman dan R. Ramdaniansyah, "Analisis Perbandingan Kinerja Query Database Management System (DBMS) antara MySQL 5.7.16 dan MARIADB 10.1," J. TEKNOIF, 2018.
- [6] V. No dan A. Junaidi, "Studi Perbandingan Performansi antara MongoDB dan MySQL dalam Lingkungan Big Data," Prosiding Annual Research Seminar 2016, 2016.
- [7] Kumar, M. S. and Jayagopal, P., "Comparison of NoSQL Database and Traditional Database-An Emphatic Analysis," JOIV Int. J. Informatics Vis., 2018.
- [8] Ganz, J., Beyer, M., and Plotzky C., "Time-series Based Solution Using InfluxDB," 2017.

- [9] Nasar, M. and Kausar, M., "Suitability of InfluxDB Database for IoT Applications," International Journal of Innovative Technology and Exploring Engineering, 2019.
- [10] Dix, P., "InfluxData (InfluxDB) Time Series Database Monitoring & Analytics," 2017.

#### BIODATA PENULIS

**Hendra, S.Kom.**, lahir di Kendari tanggal 22 Oktober, menyelesaikan pendidikan S1 pada bidang Ilmu Komputer dari STMIK AKAKOM Yogyakarta tahun 2008, dan saat ini sedang menempuh Program Magister Teknologi Informasi di STMIK AKAKOM Yogyakarta. Penulis juga aktif sebagai IT specialist Kantor Pemasaran Mandiri PT. Prudential Life Assurance di Yogyakarta.

**Dr. Widyastuti Andriyani, S.Kom., M.Kom.**, lahir di Karanganyar tanggal 17 Maret, menyelesaikan pendidikan S1 pada bidang Ilmu Komputer dari IST AKPRIND Yogyakarta tahun 2005, S2 pada bidang Ilmu Komputer dari UGM Yogyakarta tahun 2008, dan S3 pada bidang Ilmu Komputer dari UGM Yogyakarta tahun 2019. Saat ini tercatat sebagai Dosen Tetap pada Program Magister Teknologi Informasi di STMIK AKAKOM Yogyakarta dengan bidang minat kecerdasan buatan.