

Analisis kinerja sinkronisasi *database* pada DBMS *MySQL* dan *Oracle* dengan menggunakan *event-driven* dan *time-driven* untuk pemantauan data cuaca

Performance analysis of database synchronization on DBMS MySQL and Oracle using event-driven and time-driven data for weather monitoring

Nurhanif¹, Yuwaldi Away¹, dan Muhammad Syukri Surbakti^{2*},

Received 21 February 2021
Accepted 07 September 2021
Published October 2021

¹Department of Electrical Engineering & Computer, Engineering Faculty of Universitas Syiah Kuala, Darussalam Banda Aceh, 23111, Indonesia

²Physics Department, FMIPA, Universitas Syiah Kuala, Darussalam Banda Aceh, 23111, Indonesia

Abstrak. Data cuaca berupa suhu udara, kelembaban udara, curah hujan, dan kecepatan angin memerlukan proses penyimpanan yang baik supaya data cuaca dapat direkapitulasi selama satu tahun. Untuk itu membutuhkan proses penyimpanan data melalui *Database Management System* (DBMS). DBMS dapat mengelola, menyimpan sekumpulan data secara baik. Pengguna DBMS dapat membuat, mencari, memelihara, serta memiliki akses terkontrol terhadap data. Penyimpanan data pada pemantauan cuaca menggunakan *hardisk* eksternal dalam format *Comma Separated Values* (CSV), penyimpanan data seperti ini kurang efisien diakibatkan data direkapitulasi secara manual ke dalam computer. Kinerja sinkronisasi database dianalisis menggunakan metode *Event-Driven* dan *Time-Driven* pada DBMS *Oracle* dan *MySQL* untuk pemantauan data cuaca. Hasilnya, pada saat proses penyimpanan 100 data (*record*), *MySQL* memiliki kemampuan yang lebih cepat yaitu 166 detik dibandingkan dengan *Oracle* dalam waktu eksekusi 188 detik. Pada saat proses penyimpanan mulai 200 sampai 1000 data (*record*) membutuhkan waktu eksekusi rata-rata pada *Oracle* yaitu 203,56 detik sedangkan pada *MySQL* waktu eksekusi rata-rata 1163,89 detik. Jadi, kinerja DBMS *Oracle* lebih baik dibandingkan dengan DBMS *MySQL* dalam proses penyimpanan data cuaca, namun DBMS *Oracle* belum bisa menyimpan data secara langsung dari mikrokontroler karena belum adanya program yang mendukung ke dalam DBMS *Oracle*.

Abstract. Weather data in the form of air temperature, humidity, rainfall, and wind speed require a good storage process so that weather data can be recapitulated for one year. For this reason, it requires the process of storing data through a *Database Management System* (DBMS). DBMS can manage, store a collection of data well. DBMS users can create, search, maintain and have controlled access to data. Data storage in weather monitoring uses an external hard drive in *Comma Separated Values* (CSV) format, this kind of data storage is less efficient due to the data being recapitulated manually into a computer. Therefore, the author wants to analyze the performance of database synchronization on *Oracle* and *MySQL* DBMS by using *Event-Driven* and *Time-Driven* methods for monitoring weather data. As a result, when storing 100 data (*records*), *MySQL* has a faster capability of 166 seconds compared to *Oracle* in 188 seconds of execution time. When the storage process starts from 200 to 1000 data (*records*) the average execution time in *Oracle* is 203.56 seconds, while in *MySQL* the average execution time is 1163.89 seconds. So, the performance of the *Oracle* DBMS is better than the *MySQL* DBMS in the process of storing weather data, but the *Oracle* DBMS has not been able to store data directly from the microcontroller because no program supports it into the *Oracle* DBMS.

Keywords: Synchronization, DBMS *MySQL*, DBMS *MySQL*, *Time-Driven*, *Event-Driven*.

Pendahuluan

Cuaca merupakan salah satu hal yang sangat berpengaruh terhadap kehidupan makhluk hidup. Badan Meteorologi Klimatologi dan Geofisika (BMKG) bertugas untuk mengamati cuaca. Pada BMKG penyimpanan data menggunakan memori *hardisk* eksternal dalam format *Comma Separated Values* (CSV) pada batas maksimal tiga bulan. Setelah proses penyimpanan tiga bulan maka data secara otomatis terhapus untuk di simpan data tiga bulan berikutnya. Penyimpanan data seperti ini kurang efisien karena proses penyimpanan data harus dilakukan rekapitulasi secara manual ke dalam komputer.

Pengembangan yang peneliti lakukan yaitu proses penyimpanan data berbasis *Database Management*

System (DBMS). DBMS merupakan sebuah perangkat lunak yang digunakan untuk mengelola sekumpulan data yang tersimpan pada suatu *database*. Oleh karena itu, pengguna DBMS dapat membuat, mencari, memelihara, serta memiliki akses terkontrol terhadap data tersebut, seperti DBMS *MySQL* dan *Oracle* (Helinda Ayu, 2018), (Petri, 2005). *MySQL* merupakan *database open source* yang memiliki kinerja lebih cepat serta waktu eksekusi yang baik untuk digunakan sebagai penyimpanan data cuaca seperti suhu udara, kelembaban udara, curah hujan, dan kecepatan angin sebagai data lokal pada pemantauan cuaca (Geng *et al.*, 2019), (Islam *et al.*, 2017). *Oracle* adalah *database* berlisensi yang memiliki kapasitas penyimpanan data yang lebih besar serta keamanan yang

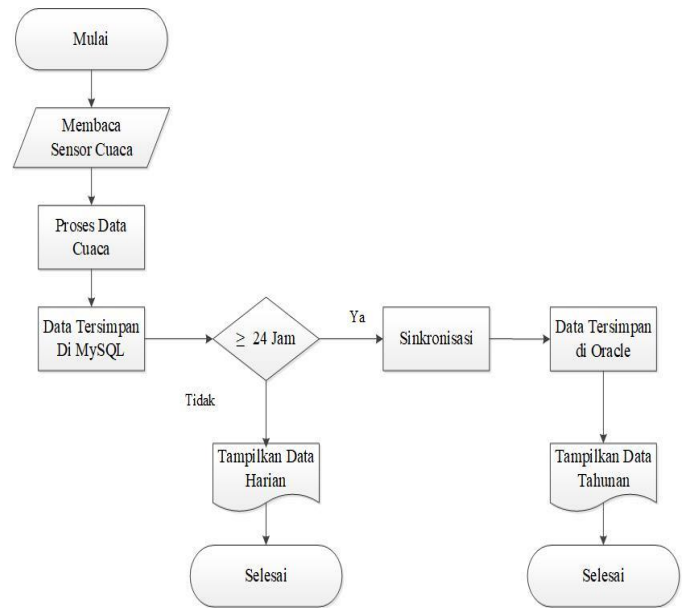
terjamin sehingga mampu menyimpan data tahunan terpusat dengan kelebihan yang dimiliki (Vaibhav, Vimal and Hiren, 2015). Selanjutnya, untuk pemantauan cuaca membutuhkan proses sinkronisasi *database* agar dapat menyimpan data lokal ke dalam data terpusat. Sinkronisasi adalah salinan data yang serupa pada *database* (Abhisena, Sukarsa and Githa, 2017). Ada banyak metode yang dapat digunakan untuk sinkronisasi data, seperti metode *Time-Driven* dan *Event-Driven*. Metode *Time-Driven* merupakan proses penyalinan data berdasarkan waktu yang digunakan untuk mengontrol dan tidak bergantung pada kejadian atau lingkungan, pergantian hari sehingga data di simpan per hari dan per minggu dengan data yang telah tersimpan di *MySQL* sebelumnya ke dalam *database Oracle* (Gani and Away, 2018), (Zou et al., 2014). Metode *Event-Driven* merupakan konsep pemrograman yang alur programnya ditentukan oleh kejadian (*event*) tertentu. Pada penelitian ini, metode *Event-Driven* digunakan untuk mengenali setiap data yang mengalami transaksi pada *database MySQL* dengan menyimpan setiap indentitas data yang mengalami perubahan kedalam sebuah tabel (Helinda Ayu, 2018).

Pada penelitian sebelumnya (Helinda Ayu, 2018), sinkronisasi dilakukan untuk *multi-platform* DBMS dengan menggunakan *Event-Driven*. Selanjutnya, proses sinkronisasi data dilakukan pada DBMS sejenis yaitu *MySQL* dan *PostgreSQL*. Kedua DBMS tersebut merupakan *database open source*. Adapun data yang digunakan pada proses pengujian sinkronisasi yaitu berupa data alokasi *bandwidth* dan media transmisi. Hasilnya, metode *Event-Driven* dapat bekerja lebih cepat dibandingkan dengan metode *Message Digest 5 (MD5)*. Pada penelitian lain (Islam et al., 2017), kinerja *database MS.SQL, Oracle, dan MySQL* dilakukan pengujian untuk proses eksekusi data. Pengujian ini dilakukan dengan menggunakan sistem operasi *windows* yang diukur dalam waktu milidetik. Hasilnya, *database MySQL* memiliki kinerja yang lebih cepat dan baik dalam melakukan proses eksekusi data namun secara umum *database* ini hanya mampu mengelola data kecil dan menengah saja (Islam et al., 2017). Berdasarkan hasil *review* tersebut belum ada yang menggunakan dua metode *Event-Driven* dan *Time-Driven* pada penyimpanan data. Oleh sebab itu penulis ingin menganalisis kinerja sinkronisasi *database* pada DBMS *Oracle* dan *MySQL* dengan menggunakan metode *Event-Driven* dan *Time-Driven* untuk pemantauan data cuaca. Selanjutnya dalam penelitian ini membahas metodologi, hasil penelitian dan kesimpulan.

Metodologi

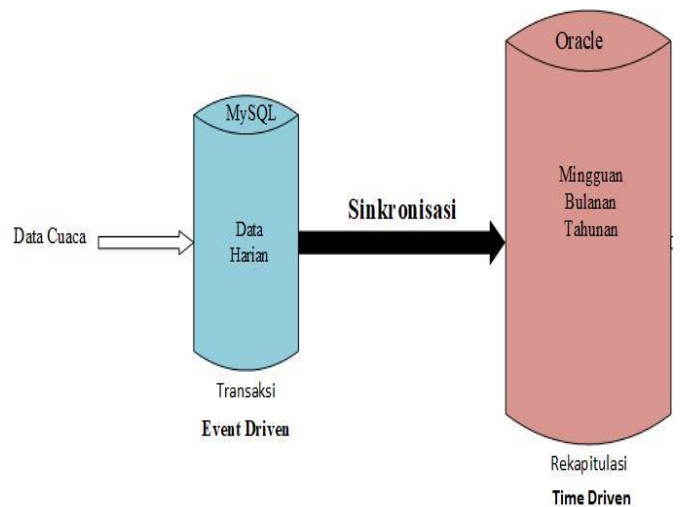
Pada penelitian ini pemantauan data cuaca dilakukan dengan menggunakan *Arduino Mega, Esp 8266, sensor suhu, kelembaban udara, curah hujan dan kecepatan angin* yang dilakukan dengan menggunakan *Event-Driven* dan *Time-Driven* pada DBMS *MySQL* dan *Oracle*. Dalam penelitian ini, penerapan *Time-Driven*

pada *Oracle* dilakukan sinkronisasi untuk pemantauan data cuaca secara harian dan mingguan berdasarkan waktu. Selanjutnya penerapan *Event-Driven* pada *MySQL* untuk merekam data berdasarkan perubahan cuaca. Jika dalam masa 24 jam maka data menampilkan sebagai data harian dan proses selesai. Selanjutnya, jika data cuaca lebih dari 24 jam maka data menyimpan di *database Oracle* sebagai data harian, mingguan, dan menampilkan pada bagian *database* dan proses selesai, dideskripsikan pada Gambar 1.



Gambar 1. Flowchart deteksi cuaca

Data cuaca yang tersimpan di dalam *database MySQL* dilakukan proses sinkronisasi ke *database Oracle* dengan menggunakan metode *Event-Driven* pada *database MySQL* dan metode *Time-Driven* pada *database Oracle*, seperti dideskripsikan pada Gambar 2.



Gambar 2. Sinkronisasi DBMS

Pada penelitian ini, tahapan untuk merancang sinkronisasi *database* terdiri dari tabel pada masing-masing DBMS. Setiap DBMS memiliki tempat penyimpanan data berupa tabel. Tabel tersebut memuat

data cuaca yang disinkronisasikan dari DBMS *MySQL* ke dalam DBMS *Oracle*.

1. Membuat tabel pada *MySQL*

Pada penelitian ini, proses pembuatan tabel dilakukan pada masing-masing DBMS. Pada DBMS *MySQL* membuat tabel dengan nama pemantauan cuaca. Struktur tabel pemantauan cuaca ditampilkan pada Tabel 1.

Tabel 1. Struktur tabel pada *MySQL*

No	Nama kolom	Tipe	Atribut
1	Nomor	Int (10)	PrimaryKey (PK)
2	Kelembaban_udara	Float	
3	Suhu	Float	
4	Curah_hujan	Float	
5	Kecepatan_angin	Float	
6	Record	Timestamp	

Untuk membuat tabel pemantauan cuaca maka perintah *query create* pada *database MySQL* adalah:
`Create table data_logger (nomor int(10), kelembaban_udara float, suhu float, curah_hujan float, kecepatan_angin float, record timestamp);`

2. Membuat tabel pada *Oracle*

Proses pembuatan tabel dilakukan pada masing-masing DBMS *Oracle* dengan nama *myora*. Pada *database Oracle* dirancang untuk menampung data dari tabel pemantauan cuaca. Struktur tabel *myora* dapat dideskripsikan pada Tabel 2.

Tabel 2. Struktur tabel pada *Oracle*

No	Nama kolom	Tipe	Atribut
1	Nomor	Int (10)	Primary Key (PK)
2	Kelembaban_udara	Float	
3	Suhu	Float	
4	Curah_hujan	Float	
5	Kecepatan_angin	Float	
6	Record	Timestamp	

Untuk membuat tabel *myora* maka perintah *query create* pada *database Oracle* terdapat perbedaan, dimana harus membuat *connection name*, *user name* dan *password*. Selanjutnya melakukan perintah *query create* dengan nama *database myora*, perintahnya adalah:
`Create table myora (nomor int(10), kelembaban_udara float, suhu float, curah_hujan float, kecepatan_angin float, record timestamp);`

3. Sinkronisasi DBMS *MySQL* dan *Oracle*

Data cuaca di simpan di dalam *database MySQL* secara *real time*. Untuk meng-*outfile* data ke dalam format CSV dengan cara melakukan *login* terlebih dahulu ke *MySQL*. Algoritma yang digunakan yaitu perintah *event scheduler*, seperti pada Algoritma 1.

2. Algoritma

2.1 Buat *Event scheduler* data *MySQL* akan menjalankan pada setiap 1 hari dimulai '2020-06-09 00:00:00'

Mulai

2.2 Pilih data kelembaban udara, suhu, curah hujan, kecepatan angin
Tulis semua data dalam file *Myora.csv*

3. Simpan semua data ke dalam *Myora.csv*
Selesai

Selanjutnya, *database MySQL* melakukan sinkronisasi dengan menggunakan perintah *event scheduler MySQL* dalam format folder CSV. Kemudian data tersebut dimasukkan ke dalam *database Oracle* dengan menggunakan *Sync Schedule Windows*. Sinkronisasi DBMS *MySQL* dan *Oracle* terdapat beberapa bagian utama yaitu program *event scheduler MySQL* yang berfungsi untuk melakukan penjadwalan pada perubahan data cuaca dan dilakukan penyimpanan dalam folder CSV.

Selanjutnya untuk meng-*input* data folder CSV memerlukan program *Load Data Oracle* yang dijalankan oleh *Sync Schedule Windows* untuk melakukan penjadwalan rekapitulasi, sehingga data tersimpan di dalam *database Oracle*, seperti pada Algoritma 2.

Algoritma 2 Penyimpanan data dalam *Oracle*

1. Membuat Load Data ke *Oracle*

2. Algoritma

2.1 Masukkan data *Myora.csv*
Ke dalam tabel *Myora* yang terdiri dari:
Nomor
Kelembaban udara
Suhu
Curah hujan
Kecepatan angin
record

2.2 Jalankan *schedule windows* pada jam 23:59 setiap hari
Untuk memasukkan data ke *Oracle*

3. Simpan data ke dalam *Oracle*
Selesai

Parameter cuaca yang dilakukan sinkronisasi ke dalam DBMS *Oracle* memiliki folder dengan format CSV. Untuk dapat meng-*input* data CSV maka digunakan *script load data Oracle* dengan nama folder yang sudah ditentukan. Selanjutnya dilakukan penjadwalan dengan menggunakan *Sync Schedule Windows*. Rekapitulasi data dilakukan dengan waktu yang ditentukan, misalnya pukul 23:59, dapat dideskripsikan dalam Gambar 3.

Algoritma 1 Pembacaan data *MySQL* ke CSV

1. Masuk ke *database MySQL*

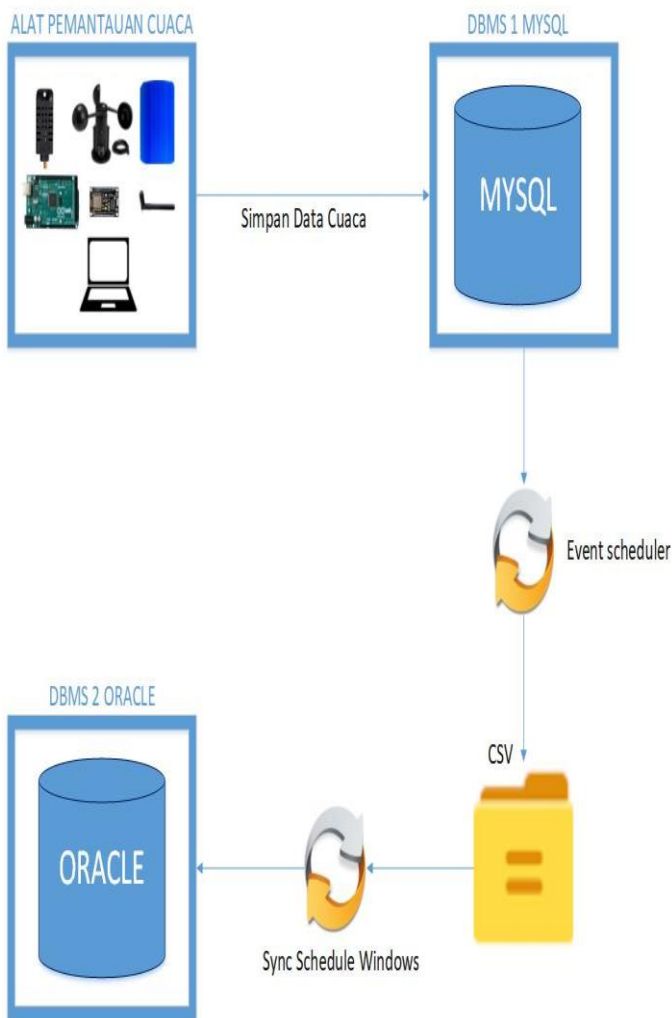
Load Data Pada Oracle

```

LOAD DATA
INFILE 'C:\xampp\mysql\data\pemantauan_cuaca\Myora.csv'
INTO TABLE MYORA
APPEND
fields terminated by ","
(
    nomor,
    suhu,
    kelembabanudara,
    curahhujan,
    kecepatanangin,
    record
)
    
```

Gambar 3. Load data pada Oracle

Arsitektur sinkronisasi DBMS pada MySQL dan Oracle selengkapnya ditunjukkan pada Gambar 4.



Gambar 4. Arsitektur Sinkronisasi

Hasil Penelitian

Dari alat pemantauan cuaca didapatkan bahwa data cuaca, selanjutnya di simpan pada DBMS MySQL. Data cuaca yang disimpan berupa suhu udara, kelembaban udara, curah hujan, kecepatan angin dan waktu record. Selengkapnya dapat dilihat pada Gambar 5.

Nomor	suhu	kelembabanudara	curah_hujan	kecepatan_angin	record
263	27.6	48.8	0.56	0.11	2020-02-29 09:05:05
264	27.6	48.8	0.56	0.11	2020-02-29 09:05:06
265	27.2	48.8	0.56	0.17	2020-02-29 09:05:07
266	27.2	48.8	0.56	0.11	2020-02-29 09:05:08
267	27.2	48.8	0.56	0.11	2020-02-29 09:05:10
268	27.4	48.8	0.56	0.11	2020-02-29 09:05:11
269	27.4	48.8	0.63	0.17	2020-02-29 09:05:12
270	27.5	48.7	0.63	0.17	2020-02-29 09:05:13
271	27.5	48.7	0.63	0.28	2020-02-29 09:05:15
272	27.4	48.8	0.63	0.23	2020-02-29 09:05:16
273	27.4	48.8	0.63	0.23	2020-02-29 09:05:17
274	27.5	48.8	0.63	0.23	2020-02-29 09:05:19
275	27.5	48.8	0.63	0.23	2020-02-29 09:05:20
276	27.4	48.8	0.63	0.23	2020-02-29 09:05:21
277	27.4	48.8	0.63	0.28	2020-02-29 09:05:23
278	27.2	48.8	0.63	0.11	2020-02-29 09:05:24
279	27.2	48.8	0.63	0.17	2020-02-29 09:05:25
280	27.3	48.7	0.63	0.17	2020-02-29 09:05:27
281	27.4	48.7	0.63	0.23	2020-02-29 09:05:28

Gambar 5. Tampilan data cuaca pada MySQL

Gambar 5 menunjukkan bahwa proses penyimpanan data cuaca dilakukan setiap hari pada DBMS MySQL, sedangkan metode Event-Driven akan melakukan sinkronisasi data setiap 1 hari setelah pukul 00:00 WIB. Secara keseluruhan, data cuaca yang tersimpan pada DBMS MySQL mengalami waktu delay selama 1 milidetik.

Selanjutnya, data cuaca berupa suhu udara, kelembaban udara, curah hujan, kecepatan angin dan waktu record juga dapat di simpan pada DBMS Oracle. Hasil selengkapnya ditunjukkan pada Gambar 6.

NOMOR	KELEMBABANUDARA	SUHU	CURAHHUIJAN	KECEPATANANGIN	RECORD
222 255	48.9	27.5	0.56	0.29-FEB-20 09.04.55.000000000 AM	
223 256	48.8	27.4	0.56	0.29-FEB-20 09.04.56.000000000 AM	
224 257	48.8	27.2	0.56	0.06 29-FEB-20 09.04.58.000000000 AM	
225 258	48.8	27.2	0.56	0.17 29-FEB-20 09.04.59.000000000 AM	
226 259	48.8	27.4	0.56	0.11 29-FEB-20 09.05.00.000000000 AM	
227 260	48.8	27.4	0.56	0.11 29-FEB-20 09.05.01.000000000 AM	
228 261	48.8	27.2	0.56	0.06 29-FEB-20 09.05.02.000000000 AM	
229 262	48.8	27.2	0.56	0.06 29-FEB-20 09.05.04.000000000 AM	
230 263	48.8	27.6	0.56	0.11 29-FEB-20 09.05.05.000000000 AM	
231 264	48.8	27.6	0.56	0.11 29-FEB-20 09.05.06.000000000 AM	
232 265	48.8	27.2	0.56	0.17 29-FEB-20 09.05.07.000000000 AM	
233 266	48.8	27.2	0.56	0.11 29-FEB-20 09.05.08.000000000 AM	
234 267	48.8	27.2	0.56	0.11 29-FEB-20 09.05.10.000000000 AM	
235 268	48.8	27.4	0.56	0.11 29-FEB-20 09.05.11.000000000 AM	
236 269	48.8	27.4	0.63	0.17 29-FEB-20 09.05.12.000000000 AM	
237 270	48.7	27.5	0.63	0.17 29-FEB-20 09.05.13.000000000 AM	
238 271	48.7	27.5	0.63	0.28 29-FEB-20 09.05.15.000000000 AM	
239 272	48.8	27.4	0.63	0.23 29-FEB-20 09.05.16.000000000 AM	
240 273	48.8	27.4	0.63	0.23 29-FEB-20 09.05.17.000000000 AM	
241 274	48.8	27.5	0.63	0.23 29-FEB-20 09.05.19.000000000 AM	
242 275	48.8	27.5	0.63	0.23 29-FEB-20 09.05.20.000000000 AM	

Gambar 6. Tampilan data cuaca pada Oracle

Gambar 6 menunjukkan bahwa penyimpanan data cuaca yang dilakukan oleh DBMS Oracle setiap harian, mingguan, bulanan dan tahunan. Data tersebut diambil dari data DBMS MySQL dengan menggunakan metode *Time-Driven*. Data yang di simpan sesuai dengan data yang tersimpan pada DBMS MySQL. Waktu yang dibutuhkan sesuai dengan jumlah data yang di simpan. Kelebihan dari DBMS Oracle adalah memiliki format waktu AM dan PM.

Pengujian penyimpanan database MySQL dan database Oracle dengan jumlah sampel data mulai dari 100, 200, 300, 400, 500, 600, 700, 800, 900 hingga 1000 data (*record*). Adapun parameter yang dilihat pada pengujian ini adalah waktu eksekusi. Hasil pengujian selengkapnya dideskripsikan pada Tabel 3.

Tabel 3. Perbandingan DBMS MySQL dengan Oracle

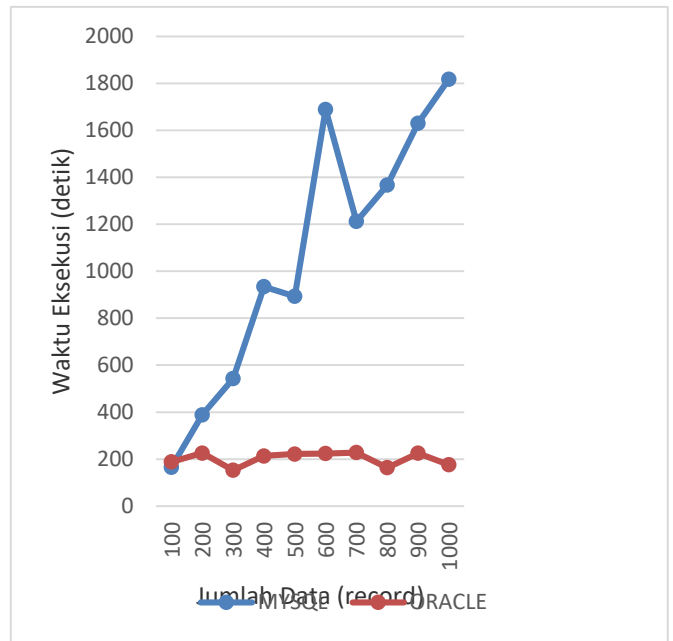
No	Jumlah Data (record)	Proses Baca MySQL (detik)	Proses Baca Oracle (detik)
1.	100	166	188
2.	200	389	226
3.	300	543	153
4.	400	934	214
5.	500	893	222
6.	600	1689	224
7.	700	1212	228
8.	800	1367	164
9.	900	1630	225
10.	1000	1818	176
Nilai Rata-rata		1064,1	202
Standar Deviasi		574,73	28,92
Modus		-	-
Min		166	153
Max		1818	228

Tabel 3 menunjukkan bahwa rata-rata waktu eksekusi yang diperlukan untuk proses sinkronisasi 1000 data (*record*) pada database MySQL adalah 1064,1 detik, sedangkan pada database Oracle adalah 202 detik.

Dari hasil pengujian juga didapatkan bahwa metode *Time-Driven* dan *Event-Driven* belum bisa digunakan pada DBMS secara bergantian, misalnya DBMS MySQL-*Event-Driven* dan DBMS Oracle - *Time-Driven* (*support*). Sedangkan DBMS MySQL-*Time-Driven* dan DBMS Oracle-*Event-Driven* (*non support*).

Waktu yang dibutuhkan pada saat penyimpanan 100 data (*record*), MySQL mampu bekerja lebih cepat dengan waktu eksekusi 166 detik sedangkan database Oracle dalam waktu 188 detik. Pada proses penyimpanan data cuaca dari 200 sampai dengan 1000 data (*record*), waktu eksekusi yang dibutuhkan oleh database Oracle lebih cepat yaitu dengan rata-rata 203,56 detik sedangkan database MySQL membutuhkan waktu eksekusi lebih lama yaitu dengan rata-rata 1163,89 detik. Perbandingan waktu eksekusi penyimpanan data dalam database

MySQL dan database Oracle dapat dilihat pada Gambar 7.



Gambar 7. Perbandingan waktu eksekusi pada penyimpanan data menggunakan DBMS MySQL dan Oracle

Berdasarkan Gambar 7 dapat diketahui bahwa waktu eksekusi untuk proses sinkronisasi data, DBMS Oracle lebih baik daripada DBMS MySQL. Untuk ruang penyimpanan data DBMS Oracle lebih banyak dibandingkan DBMS MySQL, namun DBMS Oracle juga memiliki kekurangan yaitu belum *support* untuk menyimpan data secara langsung dari mikrokontroler atau alat pemantauan cuaca.

Kesimpulan

Berdasarkan hasil penelitian, analisis kinerja sinkronisasi database pada DBMS MySQL dan Oracle dengan menggunakan metode *Event-Driven* dan *Time-Driven* untuk pemantauan data cuaca dapat disimpulkan bahwa kinerja DBMS MySQL dengan metode *Event-Driven* mampu mendapatkan data cuaca dengan akurat berdasarkan perubahan pada alat pemantauan. Sedangkan DBMS Oracle dengan menggunakan metode *Time-Driven* mampu mendapatkan data cuaca sesuai dengan data pada DBMS MySQL. DBMS Oracle lebih baik dibandingkan dengan DBMS MySQL proses sinkronisasi DBMS MySQL dan Oracle untuk pemantauan data cuaca menggunakan metode *Event-Driven* dan *Time-Driven* bekerja dengan baik. Pada saat proses penyimpanan 100 data (*record*), MySQL memiliki kemampuan yang lebih cepat yaitu 166 detik dibandingkan dengan Oracle dalam waktu eksekusi 188 detik. Pada saat proses penyimpanan mulai 200 sampai 1000 data (*record*) membutuhkan waktu eksekusi rata-rata pada Oracle yaitu 203,56 detik sedangkan pada MySQL waktu eksekusi rata-rata 1163,89 detik. Penggunaan metode *Event-Driven* dan *Time-Driven* dapat diterapkan pada proses sinkronisasi data cuaca.

Referensi

- Abhisena, G. A. Sukarsa, I. M. dan Githa, D. P. (2017) Implementasi Database Auditing dengan Memanfaatkan Sinkronisasi DBMS, Lontar Komputer. *Jurnal Ilmiah Teknologi Informasi*. **8**(2): 89.
- Gani, T. A. dan Away, Y. (2018) Pengembangan Middleware Berbasis Metode Event-Driven Untuk Sinkronisasi Database Rfid Book Drop. **3**(3): 26–31.
- Geng, D. (2019) Big data-based improved data acquisition and storage system for designing industrial data platform. *IEEE Access*. **7**: 44574–44582.
- Helinda, A. (2018) Pengembangan Algoritma untuk Sinkronisasi Multi Platform Database Management System (DBMS). *Islamic Science and Technology*. **6**(1): 109-120
- Islam, K. (2017) Huge dan Real-Time Database Systems: A Comparative Study and Review for SQL Server 2016, Oracle 12c & MySQL 5. 7 for Personal Computer'. *Journal of Basic & Applied Sciences*. **92**(21): 481–490.
- Petri, G. (2005) A Comparison of Oracle and MySQL. *Journal of Computer Science*. **2**: 41–48.
- Vaibhav, D. Vimal, V. dan Hiren, J. (2015) Comparison of Unstructured Data Handling Mechanisms in Oracle 11g and MySQL 5. 6. *Journal of Science and Technology*. **4**(1): 113–118.
- Zou, L. (2014) Time and Event Driven Communication Process for Networked Control Systems: A Survey. *Abstract and Applied Analysis*. **2014**: 1–10.