

## PENGUKURAN KINERJA STORED PROCEDURE PADA DATABASE RELASIONAL

Agus Dudu Abdulah Rohmana<sup>1</sup>, Husni Mubarak<sup>2</sup>, Rohmat Gunawan<sup>3</sup>

Jurusan Informatika Fakultas Teknik Universitas Siliwangi

Jl. Siliwangi No.24 Tasikmalaya 46115

e-mail: agusdudu14@gmail.com, husni.mubarak@unsil.ac.id, rohmatgunawan@unsil.ac.id

### Abstrak

Dalam sistem basis data, pernyataan SQL digunakan untuk menghasilkan *query* dari program *client* ke *database server*. Ini dapat memungkinkan pengguna untuk melakukan berbagai manipulasi data cepat. SQL merupakan bahasa utama yang memungkinkan server database untuk menyimpan dan mengedit data di dalamnya. Eksekusi SQL juga dapat dilakukan dengan menerapkan *Stored Procedure*. Penggunaan *stored procedure* memiliki beberapa manfaat diantaranya diantaranya waktu eksekusi *query* lebih cepat dibandingkan waktu eksekusi *query* yang disisipkan pada script bahasa pemrograman. Tujuan dari penelitian ini melakukan pengukuran waktu eksekusi dari perintah *create*, *read*, *update*, dan *delete* menggunakan *stored procedure*. Hasil percobaan pada penelitian ini, menunjukkan bahwa : waktu eksekusi *store procedure* untuk proses *create* dan *read* cenderung meningkat sesuai dengan penambahan jumlah data, tetapi untuk proses *update* dan *delete* waktu eksekusi *store procedure* cenderung bervariasi.

**Kata Kunci :** stored procedure, waktu eksekusi, query.

### Abstract

*In a database system, it is claimed that SQL is used to generate requests from client programs to the database server. This can be used by users to perform various data manipulations quickly. SQL is the main language that allows the database server to store and change the data in it. SQL execution can also be done by implementing a Stored Procedure. The stored procedure uses several questions that are published about query execution faster than the query execution time that is passed in the programming language script. The purpose of this study is to measure the execution time of commands to create, read, update, and delete using stored procedures. The results of experiments in this study, show that: the time of store procedure implementation for the process of making and reading increases according to the increase in the amount of data, but for the process of updating and removing the time of store procedure implementation can vary.*

**Keywords:** stored procedure, execution time, request.

### I. PENDAHULUAN

*Database* merupakan suatu kumpulan berbagai bentuk data yang terorganisir. *Database* juga dikenal sebagai kumpulan data terstruktur yang dapat diakses dengan berbagai cara. *Database* semakin populer karena dapat digunakan untuk mengelola data dalam jumlah besar, mudah ketika terjadi pembaruan data, serta aspek keamanan data [1]. Terdapat berbagai jenis *database*, diantaranya : *Hierarchical databases*, *Network databases*, *Relational databases*, *Object-oriented databases*, *Graph databases*, *ER model databases*, *Document databases* [2].

Database relasional memiliki beberapa manfaat yang membuatnya masih banyak digunakan saat ini: penggunaan primary key, foreign key, tipe data, aljabar relasional [3]. Fitur-fitur ini bergabung untuk membuat sistem yang konsisten dan akurat. Basis data non-relasional tidak memiliki ketiganya.

*Structured Query Language (SQL)* merupakan bahasa komputer standar untuk manajemen basis data relasional dan manipulasi data. SQL digunakan untuk query, menyisipkan, memperbarui dan memodifikasi data. Sebagian besar basis data relasional mendukung SQL, yang merupakan manfaat tambahan bagi administrator basis data [4]. SQL dieksekusi dengan

cara disisipkan pada *script* bahasa pemrograman atau menggunakan software khusus yang biasa disebut *SQL Client*.

Dalam sistem basis data, pernyataan SQL digunakan untuk menghasilkan *query* dari program *client* ke *database server*. Ini dapat memungkinkan pengguna untuk melakukan berbagai manipulasi data cepat. SQL merupakan bahasa utama yang memungkinkan server database untuk menyimpan dan mengedit data di dalamnya [5].

Eksekusi SQL juga dapat dilakukan dengan menerapkan *Stored Procedure*. *Stored Procedure* merupakan sekelompok pernyataan SQL yang telah dikompilasi sebelumnya dalam satu rencana eksekusi, dibuat dan disimpan dalam database sehingga dapat digunakan kembali ketika diperlukan [6]. Penggunaan *stored procedure* memiliki beberapa manfaat diantaranya: lebih sedikit hasil duplikasi, eksekusi lebih cepat, lalu lintas jaringan minimum dan keamanan yang lebih baik [6].

*Stored procedure* dapat digunakan untuk membagi beban *resource* yang terpakai pada saat aplikasi dijalankan. *Store procedure* juga efektif digunakan untuk eksekusi *query* yang dilakukan secara rutin diproses pada *database server* sehingga dapat memangkas waktu eksekusi [7].

Beberapa penelitian terkait pengukuran kinerja database pernah dilakukan sebelumnya, diantaranya: evaluasi kinerja waktu respons *query* dalam Document Stored NoSQL database [8], pengukuran *query* respon time pada nosql database berbasis document stored [9]. Penelitian lainnya terkait *stored procedure* diantaranya : otomatisasi SQL *query* untuk *database engine* [10], analisis perbandingan kinerja *query* Database Management System (DBMS) antara Mysql 5.7.16 dan Mariadb 10.1 [11].

Di dalam basis data *Stored Procedure* digunakan untuk mengakses data karena telah banyak memiliki keunggulan, selain meringkas penggunaan sintak *query konvensional* juga dapat meningkatkan performansi dari basis data. Sehingga lebih mudah menentukan penerapan basis data terhadap pengguna dalam menggunakan *stored procedure* pada *server*.

Berdasarkan latar belakang tersebut pada penelitian ini akan dilakukan pengukuran waktu eksekusi dari perintah *create, read, update, dan delete* menggunakan *stored procedure*.

## II. METODOLOGI

Metode yang dilakukan pada penelitian ini terdapat empat tahap yaitu; analisis sistem, identifikasi kebutuhan perangkat keras (*hardware*),

dan perangkat lunak (*software*), perancangan, implementasi dan pengujian seperti ditampilkan dalam Gambar 1.



Gambar 1. Metodologi Penelitian

### 2.1 Analisis Sistem

Pada tahap ini dilakukan analisis terhadap ruang lingkup sistem secara umum dari percobaan yang akan dikembangkan. Pengukuran waktu eksekusi *query* ini akan melibatkan *software, hardware* serta *sample data* untuk diinput pada *database*.

### 2.2 Identifikasi Kebutuhan *Hardware* dan *Software*

Pada tahap ini dilakukan identifikasi *hardware* dan *software* yang diperlukan pada proses pengukuran *query*. Spesifikasi *hardware* dan *software* yang digunakan ditampilkan pada Tabel 1 dan Tabel 2.

Tabel 1. Spesifikasi *Software* yang Digunakan

No	Software	Version
1	Microsoft Windows	10
2	MySQL Server	8.0
3	SQLyog	13.1.4

Tabel 2. Spesifikasi *Hardware* yang Digunakan

No	Item	Deskripsi
1	Processor	Intel® Core™ i5-7200U CPU @ 2.50GHz 2.71 GHz
2	Memory	8 GB
3	Graphic	Intel® HD Graphics 620

### 2.3 Perancangan

Pada tahap ini dilakukan perancangan teknis yang dilakukan pada tahap percobaan. Beberapa hal yang disiapkan sebelum dilakukan teknis proses pengukuran diantaranya :

#### a. Persiapan Data

Data yang akan dilakukan pada percobaan pada penelitian ini bertipe string. Entitas data yang diinputkan yaitu data mahasiswa dengan atribut: id, npm, nama, fakultas, jurusan, dan ipk.

#### b. Rancang Percobaan

Percobaan dilakukan dengan memproses *query* secara berulang sebanyak 500, 1000, 2500, 3000, 8000. *Query* yang dilakukan yaitu untuk proses *create, read, update, dan delete*.

c. Pengukuran

Setiap proses *query* yang dilakukan kemudian dihitung waktu *response times* yang dihasilkannya. Data hasil percobaan selanjutnya diinputkan ke dalam tabel dan disajikan dalam bentuk grafik.

## 2.4 IMPLEMENTASI DAN PENGUJIAN

Pada tahap ini dilakukan instalasi *software MySQL Server* dan *SQLyog* pada *hardware* yang telah disiapkan. Kemudian dilakukan konfigurasi pada *SQLyog* agar terkoneksi, dan proses percobaan pengukuran waktu eksekusi *stored procedure* dapat dilakukan.

## III. HASIL DAN PEMBAHASAN

### 3.1. Implementasi Query

Tahapan awal dimulai dengan pembuatan *query* untuk proses pengujian. Secara umum sintaks *query* untuk setiap proses : *create*, *read*, *update*, dan *delete* ditampilkan pada Gambar 2.

```
DELIMITER $$
USE `latihan`$$
DROP PROCEDURE IF EXISTS `sp_createdatamhs`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`sp_createdatamhs`()
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE (i<=2500) DO
        INSERT INTO mahasiswa VALUES ('1', '1270062769',
        'dudu','teknik', 'informatika', '2.0');
        SET i=i+1;
    END WHILE;
    SELECT * FROM mahasiswa;
END$$
DELIMITER ;
```

Gambar 2. Sintaks *query* untuk *create* data

Pada Gambar 2 ditampilkan *sintaks query* untuk proses *create* atau *insert* data. Pada percobaan dilakukan perulangan sebanyak 2500.

```
DELIMITER $$
USE `latihan`$$
DROP PROCEDURE IF EXISTS `sp_readmhs`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`sp_readmhs`()
BEGIN
    SELECT * FROM mahasiswa;
END$$
DELIMITER ;
```

Gambar 3. *Sintaks query* untuk *read* data

Pada Gambar 3 ditampilkan *sintaks query* untuk proses *read* atau menampilkan data.

```
DELIMITER $$
USE `latihan`$$
DROP PROCEDURE IF EXISTS `sp_updatedatamhs`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`sp_updatedatamhs`()
BEGIN
    UPDATE mahasiswa
    SET jurusan = 'sipil'
    WHERE nama = 'dudu';
END$$
DELIMITER ;
```

Gambar 4. *Sintaks query* untuk *update* data

Pada Gambar 4 ditampilkan *sintaks query* untuk proses *update* data. *Update* data dilakukan pada atribut jurusan. Data mahasiswa dengan jurusan “**Informatika**”, setelah dilakukan proses *update* berubah menjadi “**Sipil**”.

```
DELIMITER $$
USE `latihan`$$
DROP PROCEDURE IF EXISTS `sp_deldatamhs`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE
`sp_deldatamhs`()
BEGIN
    DELETE FROM `latihan`.`mahasiswa` WHERE `nama` =
    'dudu';
END$$
DELIMITER ;
```

Gambar 5. *Sintaks query* untuk *delete* data

Pada Gambar 5. ditampilkan *sintaks query* untuk proses *delete* data. *Delete* data dilakukan pada seluruh atribut yang bernama “dudu”.

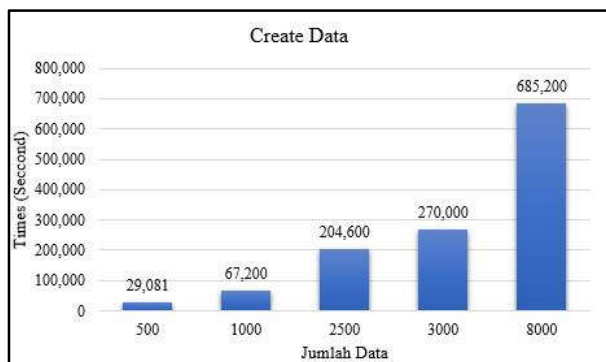
### 3.2. Perbandingan Waktu Proses Stored Procedure

*Query* yang telah dipersiapkan sebelumnya kemudian dieksekusi secara berulang. Hasil percobaan eksekusi *query* ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengukuran Waktu Proses *Stored Procedure*

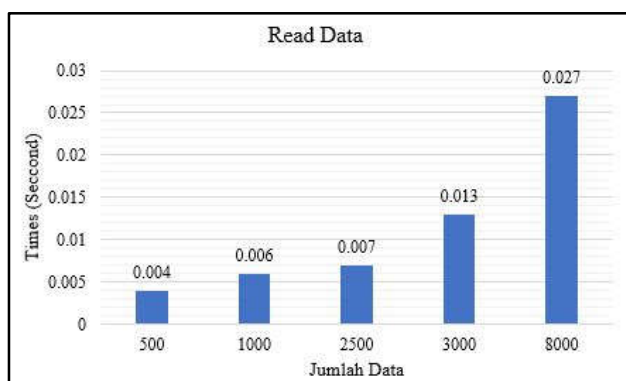
No	Jumlah Data	Query			
		Create (s)	Read (s)	Update (s)	Delete (s)
1	500	29.081	0.004	0.114	0.119
2	1.000	67.200	0.006	0.165	0.087
3	2.500	204.600	0.007	0.249	0.115
4	3.000	270.000	0.013	0.146	0.195
5	8 000	685.200	0.027	0.237	0.203

Tabel 3 menampilkan data waktu eksekusi *stored procedure* dari 5 kali percobaan untuk setiap proses *create*, *read*, *update* dan *delete*.



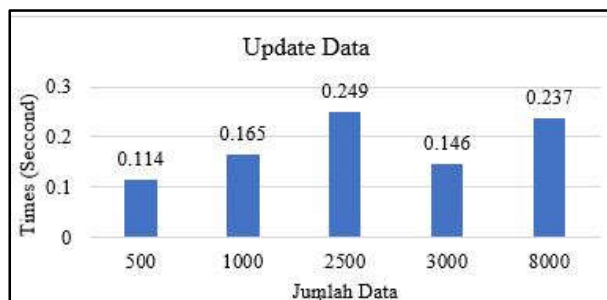
Gambar 6. Waktu eksekusi Stored Procedure *Create Data*

Pada Gambar 6 ditampilkan grafik kenaikan waktu eksekusi *stored procedure* untuk proses *create* data yang dilakukan dari 5 kali percobaan. Waktu eksekusi *stored procedure* untuk proses *create* data bertambah sesuai jumlah data yang diinputkan.



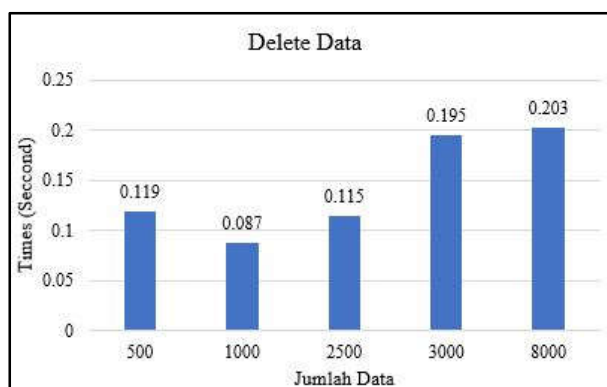
Gambar 7. Waktu eksekusi Stored Procedure *Read Data*

Pada Gambar 7 ditampilkan grafik kenaikan waktu eksekusi *stored procedure* untuk proses *read* data yang dilakukan dari 5 kali percobaan. Waktu eksekusi *stored procedure* untuk proses *read* data bertambah sesuai jumlah data yang diinputkan.



Gambar 8. Waktu eksekusi Stored Procedure *Update Data*

Pada Gambar 8 ditampilkan grafik kenaikan waktu eksekusi *stored procedure* untuk proses *update* data yang dilakukan dari 5 kali percobaan. Waktu eksekusi *stored procedure* untuk proses *update* data bertambah sesuai jumlah data yang diinputkan.



Gambar 9. Waktu eksekusi Stored Procedure *Delete Data*

Pada Gambar 9 ditampilkan grafik kenaikan waktu eksekusi *stored procedure* untuk proses *delete* data yang dilakukan dari 5 kali percobaan. Waktu eksekusi *stored procedure* untuk proses *delete* data bertambah sesuai jumlah data yang diinputkan.

#### IV. KESIMPULAN DAN SARAN

Berdasarkan hasil percobaan pada penelitian ini, diketahui bahwa : waktu eksekusi *stored procedure* untuk proses *create* dan *read* cenderung meningkat sesuai dengan pertambahan jumlah data, tetapi untuk proses *update* dan *delete* waktu eksekusi *stored procedure* cenderung bervariasi.

Perbandingan waktu eksekusi SQL dengan menerapkan *stored procedure* dan tanpa menerapkan *stored procedure*, serta penggunaan beberapa DBMS yang berbeda merupakan salah satu topik yang dapat dikembangkan pada penelitian selanjutnya.

## DAFTAR PUSTAKA

- [1] A. Akerman, "Why do we need database?," 2019. [Online]. Available: <https://www.quora.com/Why-do-we-need-database>
- [2] Arjun Panwar, "Types Of Database Management Systems," *Febrero 11*, 2019. [Online]. Available: <https://www.c-sharpcorner.com/UploadFile/65fc13/types-of-database-management-systems/>
- [3] D. Toomey, "Why are relational databases still popular when non-relational databases can give most benefits of relational DB with the flexibility of modifying needs of changing technologies?," 2019. [Online]. Available: <https://www.quora.com/Why-are-relational-databases-still-popular-when-non-relational-databases-can-give-most-benefits-of-relational-DB-with-the-flexibility-of-modifying-needs-of-changing-technologies>
- [4] Techopedia.com, "Structured query language," *Studies in Computational Intelligence*. [Online]. Available: <https://www.techopedia.com/definition/1245/structured-query-language-sql>
- [5] S. Amjal, "What Is Blockchain and How Does It Work?" [Online]. Available: <https://www.quickstart.com/blog/what-is-sql-server-and-how-does-it-work/>
- [6] Tableplus.com, "What is Stored Procedure in SQL and why should we use it?," 2019. [Online]. Available: <https://tableplus.com/blog/2018/08/what-is-stored-procedure-and-why-should-we-use-it.html>
- [7] A. D. W *et al.*, "Analisis Return on Investment (ROI) dengan Penelusuran Basis Data Guna Perhitungan Tingkat Keberhasilan Promo," *Creative Information Technology Journal*, vol. 5, no. 3, p. 196, 2019. DOI: 10.24076/citec.2018v5i3.193
- [8] R. Gunawan, "Performance Evaluation of Query Response Time in The Document Stored NoSQL Database."
- [9] R. Gunawan, "PENGUKURAN QUERY RESPON TIME PADA NOSQL DATABASE BERBASIS DOCUMENT STORED," vol. 4, no. 2, pp. 100–103, 2018.
- [10] A. Nugroho and A. S. Tohir, "Otomatisasi Sql Query Untuk Database Engine," *Semnasteknomedia Online*, vol. 5, no. 1, pp. 2–2–19, 2017 [Online]. Available: <http://webcache.googleusercontent.com/search?q=cache:http://ojs.amikom.ac.id/index.php/semnasteknomedia/article/view/1780>
- [11] I. WARMAN and R. RAMDANIANSYAH, "ANALISIS PERBANDINGAN KINERJA QUERY DATABASE MANAGEMENT SYSTEM (DBMS) ANTARA MYSQL 5.7.16 DAN MARIADB 10.1," *Jurnal Teknoif*, vol. 6, no. 1, pp. 32–41, 2018. DOI: 10.21063/jtif.2018.v6.1.32-41