

## **PERBANDINGAN PERFORMA QUERY SELECT DASAR, VIEW, DAN STORED PROCEDURE PADA DATABASE MYSQL**

### **PERFORMANCE COMPARISON OF BASIC SELECT QUERIES, VIEWS, AND STORED PROCEDURES IN MYSQL DATABASE**

**Cendana Putri Aulia<sup>1\*</sup>, Mochammad Yusuf Pratama<sup>1</sup>, Heni Lusiana Dewi<sup>1</sup>**

**\*E-mail: [cendanaputria@gmail.com](mailto:cendanaputria@gmail.com)**

<sup>1</sup>Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jawa Timur

#### **Abstrak**

Sistem informasi memerlukan tempat penyimpanan data untuk menampung data-data atau informasi di suatu organisasi. Data-data tersebut biasanya disimpan di dalam database. Dengan adanya Database Management System (DBMS), data yang ada pada database akan lebih mudah ditampilkan ataupun dikelola. Database ini menggunakan Structured Query Language (SQL) sebagai bahasa standar untuk operasi database. Pemilihan Query yang tepat untuk proses pengelolaan database sangat diperlukan untuk keefisienan, apalagi ketika menangani data dengan jumlah yang besar. Database yang dipilih adalah MySQL karena menawarkan fitur kinerja pengaksesan data yang cepat dengan jumlah data yang besar. Maka dari itu, analisis perbandingan performa query antara Select, View, dan Stored Procedure perlu dilakukan untuk mengetahui Query yang tepat. Pengujian Query dilakukan dengan mengeksekusi Query Select, View, dan Stored Procedure dengan jumlah data 500, 2500, 4500, serta 6500 data sebanyak dua kali percobaan. Hasil yang didapat menunjukkan bahwa rata-rata kecepatan performa query dari yang paling stabil, yaitu Stored Procedure, Select, dan View.

**Kata kunci:** *dbms, mysql, performance, query*

#### **Abstract**

*Information systems require a data store to host data or information in an organization. These data are usually stored in a database. With a Database Management System (DBMS), the database is easier to display or manage. It uses Structured Query Language (SQL) as the standard language for database operations. Proper Query selection for database management processes is indispensable for efficiency, especially when handling large amounts of data. The database selected is MySQL because it offers fast data access performance features with large amounts of data. Therefore, an analysis of query performance comparisons between the Select, View, and Stored Procedure needs to be done to find out the exact query. Query testing was performed by executing Query Select, View, and Stored Procedure with data amounts of 500, 2500, 4500, and 6500 data twice. Results show that the average query performance speed of the most stable is Stored Procedure, Select, and View.*

**Keywords:** *dbms, mysql, performance, query*

## **1. PENDAHULUAN**

Sistem adalah kumpulan dari komponen-komponen yang dirakit menjadi desain tertentu yang bertujuan untuk mendapatkan kegunaan yang diharapkan [1]. Komponen ini merupakan bagian yang difungsikan untuk mendukung berjalannya sistem [1]. Ada berbagai jenis sistem, yaitu sistem informasi, sistem manajemen, dan lain-lain [2][3][4]. Tujuan dari adanya sistem ini untuk

memberikan informasi, mendapat informasi, memproses data dan informasi, serta untuk mencapai tujuan dari suatu organisasi [2][3][4].

Sistem Informasi adalah sistem yang dapat digunakan untuk mengumpulkan, memproses, menyimpan, dan membagikan data atau informasi di suatu organisasi [5]. Sistem informasi ini memiliki peran yang krusial pada organisasi karena dapat memberikan informasi yang akurat serta real time untuk memfasilitasi proses pengambilan keputusan dan memungkinkan perencanaan, pengendalian, dan fungsi operasional organisasi sehingga dapat dilaksanakan secara efektif. Keberhasilan Sistem Informasi (IS) ditentukan oleh beberapa penentu utama organisasi dan lingkungan, serta keterlibatan manajemen atas maupun karyawan dalam berbagai tahap pelaksanaannya [6].

Data yang ada pada sistem informasi biasanya disimpan dalam database, yang dikelola oleh Database Management System (DBMS) [7]. Database Management System (DBMS) merupakan sebuah paket perangkat lunak yang dirancang untuk mendefinisikan, memanipulasi, mengambil, dan mengelola data dengan database [7]. DBMS ini dapat menangani tugas berorientasi transaksi (Online Transactional Processing atau OLTP), serta memungkinkan pengguna untuk mengekstrak dan mencari data secara efektif dan selektif untuk kemudian dilakukan analisis [7].

Structured Query Language (SQL) adalah bahasa pemrograman standar yang digunakan untuk mengelola basis data relasional dan melakukan operasi variabel pada data di dalamnya [7]. Dengan Query, sebuah perintah dapat dijalankan pada database untuk mengakses penyimpanan data [8]. Ketika suatu data yang berukuran besar diakses, penggunaan Query database ini perlu akses ke sumber daya DBMS dan dibutuhkan kinerja yang maksimal serta waktu yang lama untuk pemrosesan data [8]. Performa Query disini sangat penting untuk memberikan kecepatan dan ketepatan dalam mengakses data ketika jumlahnya besar. Query dapat digunakan untuk mengambil, mengolah, dan menampilkan data dari satu tabel atau beberapa tabel dengan JOIN. Selain itu, Query juga dapat menyaring data yang ingin diolah atau ditampilkan dengan WHERE. Pada DBMS juga terdapat Query SELECT, VIEW, dan STORED PROCEDURE yang digunakan untuk memproses data.

Query Select merupakan perintah yang digunakan untuk memproses pengambilan data dari tabel yang ada pada database [9]. Dengan Query Select, dapat ditentukan kolom-kolom atau tabel yang mana saja yang ingin dikelola, diakses, ataupun ditampilkan. Ada juga kondisi saat ingin menampilkan data dengan kriteria tertentu dengan filter yang disebut Where. Query Select ini akan mengambil data-data yang spesifik.

View pada DBMS merupakan tabel virtual yang berbasis pada hasil dari Query Select. View ini adalah Select yang tersimpan untuk dapat digunakan sebagai tabel pada Query berikutnya. View juga dapat digunakan untuk membuat Query yang lebih sederhana, menyembunyikan data yang sensitif untuk diakses, dan dapat memberikan abstraksi di antara pengguna dan database [10]. Selain itu, View juga dapat memberikan tampilan output data dalam database sesuai keinginan. Stored Procedure pada DBMS merupakan suatu fungsi yang ditulis terlebih dahulu untuk disimpan pada database yang nantinya akan dipanggil ketika dibutuhkan. Stored Procedure ini dapat berfungsi untuk menyingkat Query menjadi lebih sederhana. Stored Procedure juga dapat melakukan tugas, seperti memasukkan, memperbarui, menghapus, dan mengolah data menggunakan perhitungan [11]–[12]. Dengan menggunakan Stored Procedure, proses pengorganisasian logika bisnis dapat dilakukan sehingga dapat melakukan tugas yang kompleks dengan mengurangi redundansi data dan meningkatkan keefisienan.

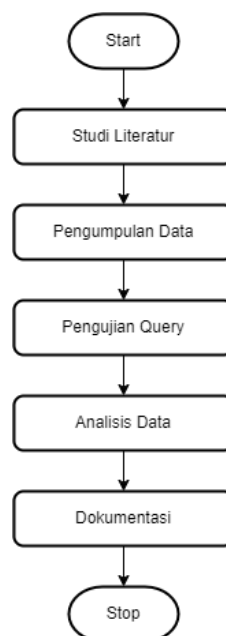
Beberapa penelitian telah dilakukan untuk menganalisis kinerja dari Query ini pada sistem DBMS. Misalnya pada sebuah studi yang menemukan bahwa Stored Procedures mampu mengungguli Function dalam waktu penyimpanan data [11]. Kemudian, ada juga studi yang menerapkan Stored Procedure untuk melihat basis DBMS saat membangun sistem informasi berbasis website untuk

tesis mahasiswa [13]. Lalu, ada juga studi yang menganalisis bagaimana kinerja dari Stored Procedure pada DBMS MYSQL [8].

Dari studi literatur tersebut, maka perlu untuk dilakukan perbandingan performa dari Query Select, View, dan Stored Procedure dalam fungsinya, yaitu untuk menyimpan dan mengolah data dalam jumlah besar. Penelitian ini dilakukan untuk mendapatkan nilai dari kecepatan pada pemrosesan data dalam jumlah tertentu yang meningkat pada satuan waktu tertentu. Uji penilaian dilakukan pada database MySQL karena MySQL menawarkan fitur-fitur dan kinerja yang baik untuk pengelolaan data dengan volume yang besar dan beban kerja yang berat dengan query yang bekerja secara cepat. Hasil yang dicari adalah nilai signifikan perbedaan waktu respon yang menunjukkan performa kecepatan dari Query Select, View, dan Stored Procedure pada database MySQL.

## 2. METODOLOGI

Metode penelitian yang dilakukan pada penelitian ini dimulai dari mencari studi literatur yang sesuai dengan penelitian ini untuk dijadikan sumber rujukan. Kemudian, dilakukan pengumpulan data. Setelah data terkumpul, dilakukan pengujian query untuk membandingkan performa masing-masing query. Selanjutnya, menganalisis data yang sudah diperoleh. Setelah itu, melakukan proses dokumentasi hasil.



**Gambar 1. Diagram Alur Metodologi**

### 2.1 Studi Literatur

Metode penelitian yang dilakukan pada penelitian ini menggunakan studi literatur. Studi literatur dilakukan untuk mengumpulkan referensi dari berbagai jurnal yang sesuai dengan penelitian yang akan dilakukan. Selanjutnya, referensi jurnal tersebut akan dijadikan acuan dalam melakukan penelitian.

### 2.2 Pengumpulan Data

Pada tahap pengumpulan data dilakukan pemilihan database yang akan digunakan pada penelitian. Penelitian ini menggunakan database sakila untuk memperoleh data yang akan digunakan dalam proses pengujian query.

### 2.3 Pengujian Query

Pada tahap pertama menguji kecepatan query select untuk menampilkan data. Pengujian dilakukan dengan batas limit 500 data yang ditunjukkan pada gambar 2, 2500 data yang ditunjukkan pada gambar 3, 4500 data yang ditunjukkan pada gambar 4, dan 6500 data yang ditunjukkan pada gambar 5. Masing-masing limit dilakukan pengulangan uji query sebanyak dua kali.

```
SELECT payment.payment_id as 'Payment ID',  
       film.title as Title,  
       customer.last_name as Customer,  
       staff.last_name as Staff,  
       payment.amount as Amount,  
       payment.payment_date as Date  
FROM payment  
JOIN customer ON payment.customer_id = customer.customer_id  
JOIN staff ON payment.staff_id = staff.staff_id  
JOIN rental ON payment.rental_id = rental.rental_id  
JOIN inventory ON rental.inventory_id = inventory.inventory_id  
JOIN film ON inventory.film_id = film.film_id  
ORDER BY payment.payment_id ASC  
LIMIT 500;
```

**Gambar 2. Query Select Limit 500**

```
SELECT payment.payment_id as 'Payment ID',  
       film.title as Title,  
       customer.last_name as Customer,  
       staff.last_name as Staff,  
       payment.amount as Amount,  
       payment.payment_date as Date  
FROM payment  
JOIN customer ON payment.customer_id = customer.customer_id  
JOIN staff ON payment.staff_id = staff.staff_id  
JOIN rental ON payment.rental_id = rental.rental_id  
JOIN inventory ON rental.inventory_id = inventory.inventory_id  
JOIN film ON inventory.film_id = film.film_id  
ORDER BY payment.payment_id ASC  
LIMIT 2500;
```

**Gambar 3. Query Select Limit 2500**

```
SELECT payment.payment_id as 'Payment ID',  
       film.title as Title,  
       customer.last_name as Customer,  
       staff.last_name as Staff,  
       payment.amount as Amount,  
       payment.payment_date as Date  
FROM payment  
JOIN customer ON payment.customer_id = customer.customer_id  
JOIN staff ON payment.staff_id = staff.staff_id  
JOIN rental ON payment.rental_id = rental.rental_id  
JOIN inventory ON rental.inventory_id = inventory.inventory_id  
JOIN film ON inventory.film_id = film.film_id  
ORDER BY payment.payment_id ASC  
LIMIT 4500;
```

**Gambar 4. Query Select Limit 4500**

```
SELECT payment.payment_id as 'Payment ID',  
       film.title as Title,  
       customer.last_name as Customer,  
       staff.last_name as Staff,  
       payment.amount as Amount,  
       payment.payment_date as Date  
FROM payment  
JOIN customer ON payment.customer_id = customer.customer_id  
JOIN staff ON payment.staff_id = staff.staff_id  
JOIN rental ON payment.rental_id = rental.rental_id  
JOIN inventory ON rental.inventory_id = inventory.inventory_id  
JOIN film ON inventory.film_id = film.film_id  
ORDER BY payment.payment_id ASC  
LIMIT 6500;
```

**Gambar 5. Query Select Limit 6500**

Pada tahap kedua menguji kecepatan query view yang disajikan pada gambar 6 untuk menampilkan data. Pengujian dilakukan dengan batas limit 500 data yang ditunjukkan pada gambar 7, 2500 data yang ditunjukkan pada gambar 8, 4500 data yang ditunjukkan pada gambar 9, dan 6500 data yang ditunjukkan pada gambar 10. Masing-masing limit dilakukan pengulangan uji query sebanyak dua kali.

```
CREATE VIEW show_payment_detail AS  
SELECT payment.payment_id as 'Payment ID',  
       film.title as Title,  
       customer.last_name as Customer,  
       staff.last_name as Staff,  
       payment.amount as Amount,  
       payment.payment_date as Date  
FROM payment  
JOIN customer ON payment.customer_id = customer.customer_id  
JOIN staff ON payment.staff_id = staff.staff_id  
JOIN rental ON payment.rental_id = rental.rental_id  
JOIN inventory ON rental.inventory_id = inventory.inventory_id  
JOIN film ON inventory.film_id = film.film_id  
ORDER BY payment.payment_id ASC
```

**Gambar 6. Query Create View**

```
select * from show_payment_detail LIMIT 500
```

**Gambar 7. Query Select View Limit 500**

```
select * from show_payment_detail LIMIT 2500
```

**Gambar 8. Query Select View Limit 2500**

```
select * from show_payment_detail LIMIT 4500
```

**Gambar 9. Query Select View Limit 4500**

```
select * from show_payment_detail LIMIT 6500
```

**Gambar 10. Query Select View Limit 6500**

Pada tahap ketiga menguji kecepatan query stored procedure yang disajikan pada gambar 11 untuk menampilkan data. Pengujian dilakukan dengan batas limit 500 data yang ditunjukkan pada gambar 12, 2500 data yang ditunjukkan pada gambar 13, 4500 data yang ditunjukkan pada gambar 14, dan 6500 data yang ditunjukkan pada gambar 15. Masing-masing limit dilakukan pengulangan uji query sebanyak dua kali.

```
CREATE PROCEDURE show_payment_detail(IN limitation INT(25))
SELECT payment.payment_id as 'Payment ID',
       film.title as Title,
       customer.last_name as Customer,
       staff.last_name as Staff,
       payment.amount as Amount,
       payment.payment_date as Date
FROM payment
JOIN customer ON payment.customer_id = customer.customer_id
JOIN staff ON payment.staff_id = staff.staff_id
JOIN rental ON payment.rental_id = rental.rental_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
ORDER BY payment.payment_id ASC
LIMIT limitation
```

**Gambar 11. Query Create Stored Procedure**

```
CALL show_payment_detail(500);
```

**Gambar 12. Query Select Stored Procedure Limit 500**

```
CALL show_payment_detail(2500);
```

**Gambar 13. Query Select Stored Procedure Limit 2500**

```
CALL show_payment_detail(4500);
```

**Gambar 14. Query Select Stored Procedure Limit 4500**

```
CALL show_payment_detail(6500);
```

**Gambar 15. Query Select Stored Procedure Limit 6500**

## 2.4 Analisis Data

Setelah melakukan tahap pengujian query, selanjutnya menganalisis data yang sudah diperoleh untuk mengetahui dan membandingkan kecepatan performa query select, view, dan stored procedure.

## 2.5 Dokumentasi

Setelah melakukan analisis data, selanjutnya hasil yang sudah diperoleh di dokumentasikan pada hasil dan pembahasan. Hasil pembahasan akan disajikan dalam bentuk tabel dan grafik disertai penjelasan yang detail.

# 3. HASIL DAN PEMBAHASAN

## 3.1 Pengujian Query Select

Pengujian pertama dilakukan dengan menguji kecepatan data pada Query Select dengan jumlah 500 data, 2500 data, 4500 data, dan 6500 data sebanyak dua kali pengujian pada perangkat komputer yang sama atau dapat disebut juga local database. Hasil dari pengujian ini dapat dilihat pada Tabel 1.

**Tabel 1. Tabel Hasil pengujian Query Select**

| S           | Jumlah Data |      |      |      |
|-------------|-------------|------|------|------|
|             | 500         | 2500 | 4500 | 6500 |
| Data 1 (ms) | 132         | 131  | 145  | 149  |

|             |     |       |     |       |
|-------------|-----|-------|-----|-------|
| Data 2 (ms) | 124 | 130   | 141 | 144   |
| Rata-rata   | 128 | 130,5 | 143 | 146,5 |

Hasil pengujian dari query Select pada tabel di atas menerangkan bahwa saat mengakses 500 data pada percobaan 1 dan 2 menghabiskan waktu 132 millisecond dan 124 millisecond dengan rata-rata waktu 128 millisecond. Data selanjutnya dengan 2500 data menghabiskan 131 millisecond pada percobaan 1 dan 130 millisecond pada percobaan 2 dengan rata-rata 130,5 millisecond. Lalu, pada percobaan 1 dan 2 dengan 4500 data menghabiskan 145 millisecond dan 141 millisecond dengan rata-rata 143 millisecond. Selanjutnya, pada percobaan 1 dan 2 dengan 6500 data menghabiskan waktu 149 millisecond dan 144 millisecond dengan rata-rata 146,5 millisecond.

### 3.2 Pengujian Query View

Pengujian kedua dilakukan dengan menguji menggunakan Query View pada data yang berjumlah 500 data, 2500 data, 4500 data, dan 6500 data pada perangkat komputer yang sama atau local database. Hasil yang didapat dapat dilihat pada tabel 2.

**Tabel 2. Tabel Hasil pengujian Query View**

| V           | Jumlah Data |       |      |      |
|-------------|-------------|-------|------|------|
|             | 500         | 2500  | 4500 | 6500 |
| Data 1 (ms) | 116         | 129   | 132  | 153  |
| Data 2 (ms) | 120         | 132   | 132  | 159  |
| Rata-rata   | 118         | 130,5 | 132  | 156  |

Pada percobaan 1 dan 2 menggunakan Query View dengan 500 data didapat hasil waktu 116 millisecond dan 120 millisecond dengan rata-rata 118 millisecond. Selanjutnya, pada percobaan 1 dan 2 dengan 2500 data menghabiskan waktu 129 millisecond dan 132 millisecond dengan rata-rata 130,5 millisecond. Percobaan selanjutnya pada 4500 data dihasilkan waktu 132 millisecond pada percobaan 1 dan 132 millisecond pada percobaan 2 dengan rata-rata 132 millisecond. Lalu, pada percobaan dengan 6500 data didapat waktu 153 millisecond pada percobaan 1 dan 159 millisecond pada percobaan 2 dengan rata-rata 156 millisecond.

### 3.3 Pengujian Query Stored Procedure

Pengujian selanjutnya dilakukan pengujian pada Query Stored Procedure dengan jumlah data yang diuji 500 data, 2500 data, 4500 data, dan 6500 data dengan menggunakan perangkat komputer yang sama atau local database. Hasil pengujian dapat dilihat pada tabel 3.

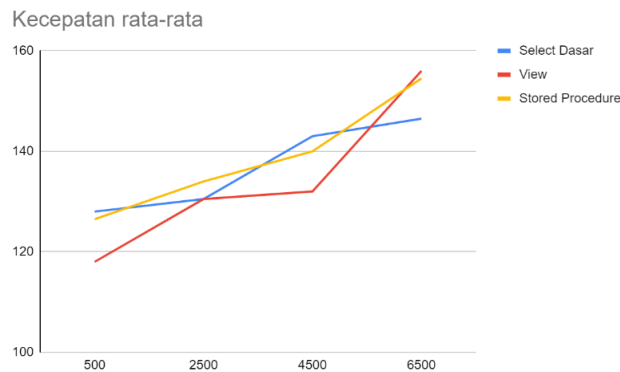
**Tabel 3. Hasil pengujian Stored Procedure**

| SP          | Jumlah Data |      |      |       |
|-------------|-------------|------|------|-------|
|             | 500         | 2500 | 4500 | 6500  |
| Data 1 (ms) | 124         | 129  | 140  | 155   |
| Data 2 (ms) | 129         | 139  | 140  | 154   |
| Rata-rata   | 126,5       | 134  | 140  | 154,5 |

Percobaan yang dilakukan pada laptop dengan Processor AMD Ryzen 3 dan AMD Radeon Graphic mendapatkan hasil percobaan pertama dan kedua dengan jumlah 500 data menghabiskan waktu 124 millisecond dan 129 millisecond dengan rata-rata 126,5 millisecond. Selanjutnya, pada percobaan dengan 2500 data mendapat hasil waktu 129 millisecond pada percobaan 1 dan 139 millisecond pada percobaan 2 dengan rata-rata 134 millisecond. Pada percobaan 1 dan 2 dengan 4500 data didapat hasil waktu 140 millisecond dan 140 millisecond dengan rata-rata 140 millisecond. Lalu, pada percobaan dengan 6500 data menghabiskan waktu 155 millisecond pada percobaan 1 dan 154 millisecond pada percobaan 2 dengan rata-rata 154,5 millisecond.



### 3.4 Kecepatan Rata-Rata



**Gambar 12. Grafik Kecepatan Rata-Rata Tiap Query**

Berdasarkan gambar 2 dapat terlihat bahwa rata-rata kecepatan performa query dari yang paling stabil yaitu stored procedure, select dasar, dan view. Kecepatan rata-rata pada query stored procedure terlihat lebih stabil meskipun dilakukan pada jumlah data yang berbeda yaitu 500, 2500, 4500, dan 6500 data. Kecepatan rata-rata pada query select dasar terlihat cukup stabil meskipun dilakukan pada jumlah data yang berbeda yaitu 500, 2500, 4500, dan 6500 data. Terlihat pada gambar grafik query select dasar kenaikannya tidak terlalu signifikan pada setiap percobaan jumlah data yang berbeda. Kecepatan rata-rata pada query view terlihat kurang stabil dilakukan pada jumlah data yang berbeda yaitu 500, 2500, 4500, dan 6500 data. Terlihat pada gambar grafik query view kenaikannya terlalu signifikan pada setiap percobaan jumlah data yang berbeda.

### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa perintah yang memiliki rata-rata kecepatan respon query dari yang paling stabil, yaitu Stored Procedure, Select dasar, dan View. Namun, setiap penggunaan dari ketiga perintah tersebut memiliki fungsinya masing-masing. Penggunaan query Select dasar, View, maupun Stored Procedure memiliki alasannya masing-masing pada sebuah sistem yang dijalankan dan tidak ada suatu keharusan pada suatu sistem untuk menggunakan query tertentu.

### 5. DAFTAR RUJUKAN

- [1] Sudarno, R. Santoso, dan A. Anugraheni, "RELIABILITAS DAN AVAILABILITAS SISTEM TIGA KOMPONEN TERSUSUN PARALEL BERSERI," *Statistika*, vol. 6, no. 2, hlm. 130–136, Nov 2018, [Daring]. Tersedia pada: <http://jurnal.unimus.ac.id>
- [2] Suhari, A. Faqih, dan F. M. Basysyar, "Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. Angkasa Raya," *Jurnal Teknologi dan Informasi (JATI)*, vol. 12, no. 1, hlm. 30–45, Mar 2022, doi: 10.34010/jati.v12i1.
- [3] D. Apriani, T. Ramadhan, E. Astriyani, Mulyati, dan Mardiana, "Kerja Lapangan Berbasis Website Untuk Sistem Informasi Manajemen Praktek (Studi Sistem Informasi Program Studi Kasus Merdeka Belajar Kampus Merdeka (MBKM) Universitas Raharja," *ADI Bisnis Digital Interdisiplin Jurnal*, hlm. 23–29, 2022.
- [4] A. Mayasari, Y. Supriani, dan O. Arifudin, "Implementasi Sistem Informasi Manajemen Akademik Berbasis Teknologi Informasi dalam Meningkatkan," *JHIP-Jurnal Ilmiah Ilmu Pendidikan*, vol. 4, no. 5, hlm. 340–345, Sep 2021, [Daring]. Tersedia pada: <http://Jiip.stkipyapisdompu.ac.id>



- [5] Порван, А. Павлович, dan Ю. В. Журавлева, "Information system definition of adaptation opportunities of the students organism," *Technology audit and production reserves* 6, hlm. 38–41, 2013, [Daring]. Tersedia pada: <http://annamed>.
- [6] M. Ghobakhloo dan S. H. Tang, "Information system success among manufacturing SMEs: case of developing countries," *Inf Technol Dev*, vol. 21, no. 4, hlm. 573–600, Okt 2015, doi: 10.1080/02681102.2014.996201.
- [7] P. V. Openko, S. Yu. Hohoniants, O. V. Starkova, K. V. Herasymenko, M. I. Yastrebov, dan A. O. Prudchenko, "Problem of Choosing a DBMS in Modern Information System," dalam *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, IEEE, Des 2019, hlm. 171–174. doi: 10.1109/ATIT49449.2019.9030517.
- [8] I. Warman dan Wildani, "ANALISA KINERJA QUERY STORED PROCEDURE PADA DATABASE MANAGEMENT SYSTEM (DBMS) MYSQL," *Jurnal Sains dan Teknologi*, vol. 21, no. 1, hlm. 58–63, Jun 2021, [Daring]. Tersedia pada: <https://db-engines.com>,
- [9] A. H. Fathulloh dan H. I. Adauwiyah, "Perbandingan Tingkat Efisiensi Waktu Query SELECT pada Database Interface Navicat dan SQLYog di MySQL DBMS," *Applied Information System and Management (AISM)*, vol. 4, no. 2, hlm. 101–105, Okt 2021, doi: 10.15408/aism.v4i2.18369.
- [10] Y. Han, C. Chai, J. Liu, G. Li, C. Wei, dan C. Zhan, "Dynamic Materialized View Management using Graph Neural Network," 2022.
- [11] A. D. Septiadi dan L. Jeong-Bae, "Comparative Analysis of Database Query Storage Performance Between Stored Procedure and Function," *International Journal of Informatics and Information System*, vol. 3, no. 2, hlm. 60–66, 2020.
- [12] N. Hartono dan Erfina, "Comparison of Stored Procedures on Relational Database Management Systems," *JOURNAL TECH-E*, vol. 4, no. 2, hlm. 8–15, 2021, [Daring]. Tersedia pada: <http://bsti.ubd.ac.id/e-jurnal>
- [13] A. N. Am, "Implementasi Stored Procedure Dan View Basis DBMS Dalam Membangun Sistem Informasi Tugas Akhir Mahasiswa Politeknik Kampar," *Jurnal Sains dan Ilmu Terapan*, vol. 3, no. 2, hlm. 2621–766, 2020, [Daring]. Tersedia pada: <http://www.php.net>.