# IMPLEMENTASI TEKNIK QUERY OPTIMIZATION UNTUK MENINGKATKAN PERFORMA SQL SERVER PADA SISTEM INFORMASI AKADEMIK

**Salam Patulus Rahmat Tambunan, MHD Arief Hasan*, Rezky Hartono Tambunan, Michael Stephen Siahaan, Michael Jordan Sirait, Farid Mahmud**
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Lancang Kuning
Jl. Yos Sudarso KM. 8 Rumbai, Pekanbaru, Riau
*m.arif@unilak.ac.id*

**ABSTRAK**

Sistem Informasi Akademik (SIAKAD) berperan penting dalam pengelolaan data akademik di perguruan tinggi, terutama Kartu Rencana Studi (KRS) mahasiswa. Namun, meningkatnya jumlah data dan pengguna sering menyebabkan performa sistem menurun akibat eksekusi query SQL yang kurang efisien. Masalah ini berdampak pada lambatnya proses data, konflik jadwal, dan akses informasi. Penelitian ini bertujuan mengoptimalkan performa SQL server melalui teknik seperti pengindeksan dan penggunaan stored procedures. Hasil implementasi menunjukkan peningkatan signifikan pada waktu eksekusi query dan pengurangan beban server. Pengujian lima kali menunjukkan performa yang lebih baik. Penggunaan indexing menurunkan rata-rata penggunaan CPU dari 9,4 menjadi 3, serta waktu eksekusi query dari rata-rata 249 ms menjadi 129 ms—peningkatan 93%. Sementara itu, stored procedures meski meningkatkan rata-rata penggunaan CPU dari 6 menjadi 9,4, tetap memberikan pengurangan waktu eksekusi rata-rata menjadi 117,6 ms, dengan peningkatan performa hingga 95%. Optimasi ini menjadikan SIAKAD lebih responsif dan efisien, mendukung pengelolaan data akademik secara cepat, akurat, dan handal, sehingga meningkatkan pengalaman pengguna, termasuk mahasiswa, dosen, dan pihak administrasi.

***Kata kunci :*** *Pengoptimalan Query, SQL Server, Sistem Informasi Akademik, Performa Basis Data, Pengindeksan, Prosedur Tersimpan*

## 1. PENDAHULUAN

Di era perkembangan informasi yang semakin pesat, pengelolaan data akademik di perguruan tinggi menjadi aspek krusial yang memerlukan perhatian serius. Sistem Informasi Akademik (SIAKAD) berfungsi sebagai pusat utama dalam mengelola berbagai data akademik, seperti penjadwalan perkuliahan, pengisian Kartu Rencana Studi (KRS), serta penyimpanan dan pengolahan informasi mahasiswa [1].

Seiring meningkatnya jumlah mahasiswa dan semakin kompleksnya data yang harus dikelola, SIAKAD dituntut untuk beroperasi secara efisien dan responsif. Namun, tantangan utama yang sering muncul adalah performa sistem, khususnya terkait kecepatan dan efisiensi eksekusi query SQL [2].

Query SQL yang belum dioptimalkan dapat menimbulkan beberapa permasalahan, seperti waktu pemrosesan data yang lambat, peningkatan beban server, dan keterlambatan dalam mengakses informasi [3].

Hal ini berdampak langsung pada pengalaman pengguna, terutama mahasiswa yang ingin mengakses jadwal perkuliahan atau melakukan pengisian KRS. Kondisi ini berpotensi menghambat aktivitas operasional akademik dan mengganggu kelancaran proses belajar mengajar. Oleh sebab itu, penerapan teknik optimasi query menjadi langkah penting untuk meningkatkan performa SQL Server dalam mendukung operasional SIAKAD.

Optimasi query bertujuan untuk meningkatkan efisiensi eksekusi perintah SQL dengan meminimalkan waktu pengambilan data dari database [4].

Beberapa teknik yang dapat diterapkan meliputi penggunaan indeks, penulisan ulang query, serta implementasi stored procedures. Teknik pengindeksan, misalnya, dapat mempercepat proses pencarian data dengan menyediakan struktur tambahan yang memudahkan akses lebih cepat. Sementara itu, teknik penulisan ulang query membantu menyederhanakan perintah SQL sehingga proses eksekusi menjadi lebih efisien. Penggunaan stored procedures juga berperan penting dalam mengurangi beban server dengan membatasi jumlah data yang dikirim antara aplikasi dan database.

Penelitian sebelumnya oleh Himawan dan Legia (2018) berfokus pada pengembangan Sistem Informasi Akademik (SIAKAD) untuk mendukung penjadwalan mata kuliah dan pengisian Kartu Rencana Studi (KRS) di Universitas Matana. Penelitian ini menggunakan pendekatan SQL optimizer untuk mengurangi benturan jadwal dan meningkatkan efisiensi pengelolaan jadwal akademik. Namun, penelitian tersebut memiliki keterbatasan, yaitu cakupannya yang hanya berfokus pada aspek penjadwalan mata kuliah dan kurang memberikan perhatian pada efisiensi keseluruhan performa sistem. Selain itu, penelitian tersebut hanya dirancang untuk kebutuhan spesifik Universitas Matana tanpa mempertimbangkan fleksibilitas atau skalabilitas yang memungkinkan penerapan di institusi lain [3].

Sebaliknya, dengan adanya penelitian ini, diharapkan dapat memperluas cakupan penelitian dengan mengintegrasikan teknik optimasi query untuk

meningkatkan performa keseluruhan SQL Server dalam SIAKAD. Pendekatan yang digunakan dalam penelitian ini mencakup teknik indexing dan stored procedures, yang dirancang untuk mempercepat eksekusi query, mengurangi beban server, dan meningkatkan pengalaman pengguna, termasuk mahasiswa, dosen, dan pihak administrasi. Selain itu, penelitian ini menekankan pentingnya fleksibilitas dan responsivitas sistem, menjadikannya relevan tidak hanya bagi perguruan tinggi tertentu tetapi juga untuk institusi lain dengan tantangan serupa.

Salah satu kesenjangan utama yang diisi oleh penelitian ini adalah penerapan teknik spesifik yang terbukti secara kuantitatif meningkatkan performa sistem. Dalam penelitian sebelumnya, efisiensi sistem hanya diukur dari aspek pengelolaan jadwal akademik, sementara penelitian saat ini menunjukkan peningkatan performa secara menyeluruh, seperti pengurangan durasi eksekusi query. Selain itu, penggunaan alat seperti SQL Profiler memberikan pendekatan yang lebih sistematis dalam menganalisis dan mengoptimalkan performa database.

Penelitian ini juga mengisi celah dalam hal fleksibilitas implementasi. Jika penelitian sebelumnya dirancang secara spesifik untuk kebutuhan Universitas Matana, penelitian ini dirancang agar dapat diterapkan di berbagai institusi pendidikan yang menghadapi tantangan serupa dalam pengelolaan data akademik. Dengan hasil yang lebih komprehensif dan aplikatif, penelitian ini memberikan kontribusi signifikan terhadap pengembangan SIAKAD yang responsif, efisien, dan adaptif terhadap kebutuhan perguruan tinggi modern.

## 2. TINJAUAN PUSTAKA

### 2.1. Pengoptimalan Query

Pengoptimalan Query adalah suatu proses untuk menganalisa query untuk menentukan sumber-sumber apa saja yang digunakan oleh query tersebut dan apakah penggunaan dari sumber tersebut dapat dikurangi tanpa merubah output. Atau bisa juga dikatakan bahwa pengoptimalan query adalah sebuah prosedur untuk meningkatkan strategi evaluasi dari suatu query untuk membuat evaluasi tersebut menjadi lebih efektif. Pengoptimalan query mencakup beberapa teknik seperti transformasi query ke dalam bentuk logika yang sama, memilih jalan akses yang optimal dan mengoptimumkan penyimpanan data [7].

Pengoptimalan query dapat membantu sistem informasi dalam memproses data lebih cepat, meningkatkan performa server, dan memperbaiki pengalaman pengguna. Teknik optimasi ini sangat relevan untuk Sistem Informasi Akademik (SIAKAD) yang membutuhkan pengolahan data yang cepat dan efisien.

### 2.2. Indexing

Mengindeks kolom dalam tabel adalah cara yang bisa dilakukan untuk mengoptimalkan hasil pencarian query SQL. Membuat indeks dalam sql tak ubahnya seperti membuat daftar indeks pada buku. Penggunaan indeks pada eksekusi query SQL akan benar-benar terasa ketika tabel dan basis data yang kita punya sudah cukup besar [7].

Dalam konteks penelitian ini, teknik pengindeksan diterapkan untuk mempercepat eksekusi query pada SIAKAD. Dengan mengindeks kolom yang sering digunakan dalam operasi filter atau join, durasi pencarian data dapat dikurangi, sehingga performa sistem menjadi lebih responsif.

### 2.3. Stored Procedures

Stored Procedures 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 [5] [6].

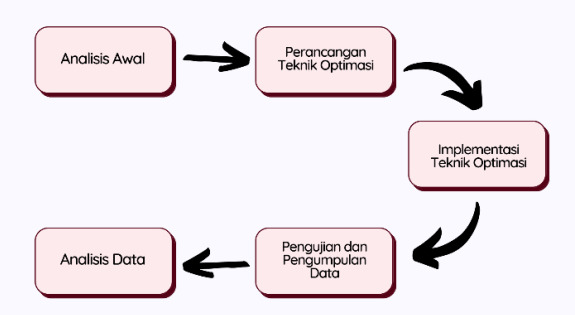
Pada Penelitian ini menggunakan stored procedures untuk mengoptimalkan operasi basis data pada SIAKAD. Dengan meminimalkan kompleksitas query dan memaksimalkan efisiensi eksekusi, stored procedures mampu memberikan pengurangan waktu eksekusi query secara signifikan.

### 2.4. SQL Server Profiler

SQL Server Profiler adalah alat bawaan dalam SQL Server yang digunakan untuk memantau dan melacak aktivitas database, termasuk performa query. alat ini memungkinkan pengguna untuk merekam jejak (trace) peristiwa yang terjadi dalam instance SQL Server, yang sangat membantu dalam mengidentifikasi masalah performa, debugging query, dan mengoptimalkan sistem [8].

Dalam penelitian ini, SQL Server Profiler digunakan untuk menganalisis performa query sebelum dan sesudah teknik optimasi diterapkan. Analisis ini memberikan data kuantitatif yang mendukung evaluasi keberhasilan implementasi metode optimasi, seperti indexing dan stored procedures.

## 3. METODE PENELITIAN




Gambar 1. Alur Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang sistematis untuk memastikan keberhasilan implementasi teknik optimasi query

dalam meningkatkan performa SQL Server pada Sistem Informasi Akademik.

### 3.1. Analisis Awal

Pada tahap ini, langkah pertama yang dilakukan adalah identifikasi masalah. Peneliti menganalisis query-query yang sering digunakan dalam Sistem Informasi Akademik untuk menemukan query yang menjadi penyebab lambatnya proses eksekusi atau bottleneck pada performa sistem. Identifikasi ini dilakukan dengan menganalisis log dan statistik penggunaan query. Selanjutnya, dilakukan pengukuran performa awal untuk mendapatkan baseline performa sebelum optimasi diterapkan. Pengukuran ini meliputi pencatatan waktu eksekusi query, penggunaan CPU, konsumsi memori, dan aktivitas I/O disk. Tools seperti SQL Profiler, Query Store, dan Performance Monitor digunakan untuk memperoleh data performa yang akurat.

### 3.2. Perancangan Teknik Optimasi

Setelah masalah dan performa awal teridentifikasi, peneliti merancang teknik optimasi yang akan diterapkan. Tahap ini diawali dengan pemilihan teknik query optimization yang relevan dengan permasalahan yang ditemukan. Teknik yang dirancang meliputi:

a. Indexing, yaitu pembuatan indeks pada kolom yang sering digunakan dalam filter atau join untuk mempercepat proses pencarian data.
b. Stored Procedures, yaitu teknik untuk mengurangi beban server dengan membatasi jumlah data yang dikirim antara aplikasi dan database.

Setelah teknik optimasi dipilih, dilakukan desain eksperimen untuk menentukan query-query yang akan dioptimasi dan menyusun baseline pengukuran performa sebelum implementasi. Hal ini bertujuan untuk memastikan evaluasi performa dilakukan secara sistematis.

### 3.3. Implementasi Teknik Optimasi

Pada tahap ini, teknik-teknik optimasi yang telah dirancang diterapkan pada SQL Server. Setiap perubahan yang dilakukan, baik pada struktur database maupun query, didokumentasikan secara rinci. Dokumentasi ini mencakup deskripsi perubahan, hasil pengujian awal setelah implementasi, dan alasan di balik penerapan teknik tersebut.

### 3.4. Pengujian dan Pengumpulan Data

Setelah implementasi selesai, dilakukan uji coba sistem untuk mengukur dampak teknik optimasi yang telah diterapkan. Uji coba dilakukan dengan menjalankan query pada database yang telah dioptimasi menggunakan data transaksi nyata dari Sistem Informasi Akademik. Hasil uji coba digunakan untuk pengumpulan data performa sistem setelah optimasi. Data yang dicatat meliputi waktu eksekusi query, penggunaan CPU, konsumsi memori, dan aktivitas I/O disk. Data ini dibandingkan dengan baseline yang telah disusun sebelumnya untuk mengukur peningkatan performa secara kuantitatif.

### 3.5. Analisis Data

Tahap terakhir adalah analisis data untuk mengevaluasi efektivitas teknik optimasi yang diterapkan. Analisis performa dilakukan dengan membandingkan data sebelum dan sesudah optimasi untuk mengetahui sejauh mana peningkatan yang berhasil dicapai. Analisis ini juga menggunakan metode statistik untuk memberikan evaluasi yang lebih objektif terhadap hasil penelitian. Selain itu, dilakukan evaluasi teknik optimasi untuk menentukan teknik mana yang memberikan peningkatan performa paling signifikan. Jika ada query yang tetap tidak optimal, peneliti akan menyelidiki penyebabnya dan memberikan saran untuk perbaikan lebih lanjut.

## 4. HASIL DAN PEMBAHASAN

### 4.1. Analisis Awal

Langkah awal yang dilakukan adalah melakukan analisis kueri yang sering digunakan dalam SIAKAD. Dalam SIAKAD, ada beberapa kueri yang umum digunakan untuk menangani berbagai operasi yang berkaitan dengan administrasi mahasiswa, salah satunya adalah

```
SELECT * FROM mahasiswa WHERE prodi = 'Teknik Informatika';
```

Gambar 2. Analisis Awal

Kueri ini digunakan untuk mencari mahasiswa dengan prodi teknik informatika. Kueri pencarian ini umum digunakan untuk monitoring performa akademik atau evaluasi jumlah mahasiswa aktif di setiap program studi. Kueri ini juga menjadi pondasi awal untuk melakukan kueri yang lebih kompleks lagi.

### 4.2. Pembuatan Indexing

Teknik optimasi indexing meningkatkan efisiensi pencarian data dalam basis data dengan memanfaatkan indeks secara optimal. Langkah utama meliputi analisis pola kueri, pembuatan indeks pada kolom yang sering digunakan, dan penerapan indeks gabungan untuk operasi kompleks. Dengan implementasi yang tepat, optimasi indeks dapat mempercepat kueri secara signifikan.

```
CREATE INDEX idx_prodi ON mahasiswa
```

Gambar 3. Pembuatan Indexing

### 4.3. Pembuatan Stored Procedure

Teknik optimasi stored procedures bertujuan meningkatkan kinerja, keamanan, dan efisiensi basis data dengan cara memanfaatkan prosedur tersimpan (stored procedures) secara optimal. Stored procedures

adalah sekumpulan perintah SQL yang disimpan dan dijalankan di sisi server database, sehingga mengurangi lalu lintas jaringan antara aplikasi dan server. Implementasinya dimulai dengan mengidentifikasi operasi yang sering dilakukan, seperti pengolahan data kompleks atau kueri berulang, kemudian mengemas dalam stored procedures.

```
-- Membuat prosedur

CREATE PROCEDURE cari_mahasiswa_by_prodi
    @prodi VARCHAR(50) -- Parameter input
AS
BEGIN
    SELECT nim, nama, prodi
    FROM mahasiswa
    WHERE prodi = @prodi
    ORDER BY nama;
END;
```

Gambar 4. Pembuatan prosedur

### 4.4. Dataset

Penelitian ini menggunakan data dummy yang dibuat di SQL Server Profiler. Data tersebut terdiri dari 5 tabel yaitu tabel dosen, tabel jadwal kuliah, tabel krs, tabel mahasiswa dan tabel mata kuliah. Tabel dosen memiliki total data 500, tabel jadwal kuliah memiliki total data 1000, tabel krs memiliki total data 10000, tabel mahasiswa memiliki total data 10000, dan tabel mata kuliah memiliki total data 50 data.

### 4.5. Pengujian Teknik Optimasi

Pengujian ini dilakukan sebanyak 5 kali pengujian pada kueri pencarian mahasiswa dengan program studi teknik informatika.

```
SELECT nim, nama, prodi
FROM mahasiswa
WHERE prodi = 'Teknik Informatika';
```

Gambar 5. Kueri pencarian untuk pengujian

Pengujian pertama dilakukan dengan menggunakan metode indexing. Hasil ini didapatkan berdasarkan analisis menggunakan SQL Server Profiler yang ditunjukkan pada tabel ini:

Tabel 1. Hasil pengujian sebelum menggunakan Indexing

| Pengujian | CPU | Reads | Writes | Duration |
|---|---|---|---|---|
| 1 | 16 | 84 | 0 | 357 |
| 2 | 31 | 82 | 0 | 198 |
| 3 | 0 | 82 | 0 | 198 |
| 4 | 0 | 82 | 0 | 259 |
| 5 | 0 | 82 | 0 | 233 |
| Rerata | 9.4 | 82.4 | 0 | 249 |

Tabel 2. Hasil pengujian setelah menggunakan Indexing

| Pengujian | CPU | Reads | Writes | Duration |
|---|---|---|---|---|
| 1 | 0 | 86 | 0 | 146 |
| 2 | 0 | 82 | 0 | 106 |
| 3 | 15 | 82 | 0 | 125 |
| 4 | 0 | 82 | 0 | 149 |
| 5 | 0 | 82 | 0 | 119 |
| Rerata | 3 | 82.8 | 0 | 129 |

Pada pengujian teknik indexing, hasil menunjukkan adanya peningkatan signifikan dalam performa sistem setelah indeks diterapkan. Sebelum menggunakan indexing, rata-rata durasi eksekusi query berada pada 249 ms dengan penggunaan CPU rata-rata sebesar 9,4%. Durasi eksekusi tertinggi mencapai 357 ms, yang menunjukkan inefisiensi dalam pengolahan query. Namun, setelah penerapan indexing, rata-rata durasi eksekusi berkurang drastis menjadi 129 ms, dengan penggunaan CPU menurun hingga 3%. Peningkatan ini mencerminkan efisiensi yang lebih tinggi dalam pengelolaan data, terutama pada pencarian dalam tabel besar. Efektivitas indexing terlihat dari peningkatan performa hingga 93%, membuat teknik ini sangat cocok untuk mempercepat query sederhana yang sering digunakan, seperti operasi filter dan join. Pengujian selanjutnya menggunakan metode stored procedures. Hasil ini didapatkan berdasarkan analisis menggunakan SQL Server Profiler yang ditunjukkan pada tabel ini:

Tabel 3. Hasil pengujian sebelum menggunakan Stored Procedures

| Pengujian | CPU | Reads | Writes | Duration |
|---|---|---|---|---|
| 1 | 0 | 82 | 0 | 423 |
| 2 | 0 | 82 | 0 | 195 |
| 3 | 0 | 82 | 0 | 188 |
| 4 | 15 | 132 | 0 | 188 |
| 5 | 15 | 132 | 0 | 158 |
| Rerata | 6 | 102 | 0 | 230.4 |

Tabel 4. Hasil pengujian menggunakan Stored Procedures

| Pengujian | CPU | Reads | Writes | Duration |
|---|---|---|---|---|
| 1 | 16 | 82 | 0 | 108 |
| 2 | 16 | 82 | 0 | 130 |
| 3 | 15 | 82 | 0 | 95 |
| 4 | 0 | 82 | 0 | 101 |
| 5 | 0 | 82 | 0 | 154 |
| Rerata | 9.4 | 82 | 0 | 117.6 |

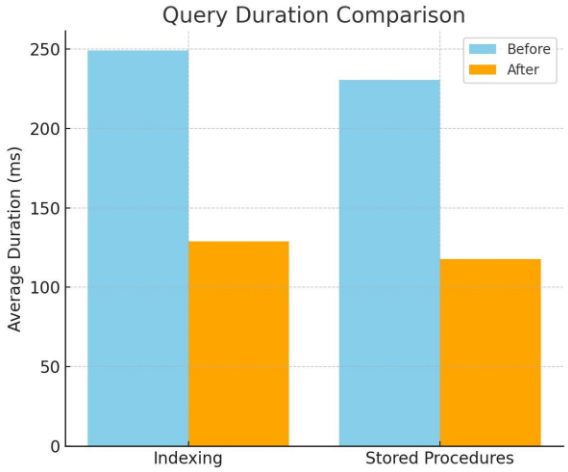
Pada pengujian dengan stored procedures, performa juga meningkat secara signifikan, meskipun dengan pola yang sedikit berbeda. Sebelum penerapan stored procedures, rata-rata durasi eksekusi query adalah 230,4 ms dengan penggunaan CPU sebesar 6%. Setelah implementasi, durasi eksekusi berkurang menjadi rata-rata 117,6 ms, menunjukkan peningkatan performa hingga 95%. Namun, penggunaan CPU mengalami peningkatan menjadi rata-rata 9,4%. Hal ini menunjukkan bahwa stored procedures efektif

untuk mengurangi durasi query yang kompleks dengan mengalihkan sebagian besar pengolahan ke sisi server, meskipun membutuhkan sumber daya CPU yang sedikit lebih tinggi.

Secara keseluruhan, baik teknik indexing maupun stored procedures memiliki keunggulan masing-masing. Indexing sangat efektif untuk mengurangi beban CPU dan mempercepat pencarian data dalam tabel besar, sedangkan stored procedures unggul dalam menangani query yang kompleks dengan durasi eksekusi lebih pendek. Kombinasi kedua teknik ini dapat memberikan hasil optimal dalam meningkatkan performa SQL Server pada Sistem Informasi Akademik (SIAKAD), menjadikannya lebih responsif dan efisien dalam mendukung kebutuhan pengguna.

### 4.6. Grafik Performa Optimalisasi

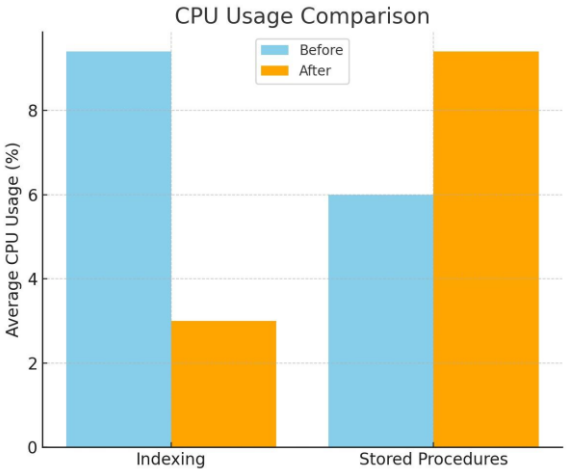
Performa metode indexing dan metode stored procedures dapat dilihat berdasarkan grafik performa ini




Gambar 6. Grafik performa terhadap durasi eksekusi

Grafik performa durasi eksekusi menunjukkan tren penurunan signifikan setelah penerapan teknik indexing dan stored procedures. Sebelum optimasi, durasi eksekusi query rata-rata cukup tinggi, dengan angka tertinggi mencapai 357 ms pada skenario tanpa optimasi. Setelah penerapan indexing, durasi eksekusi rata-rata menurun drastis hingga 129 ms, dengan penurunan tertinggi terlihat pada query dengan durasi eksekusi awal yang lebih lama. Teknik ini memberikan peningkatan efisiensi sebesar 93% pada pengolahan query sederhana.

Sementara itu, penggunaan stored procedures juga menunjukkan penurunan durasi eksekusi yang signifikan, dari rata-rata 230,4 ms menjadi 117,6 ms. Penurunan ini bahkan lebih besar, mencapai 95%, dibandingkan dengan teknik indexing. Grafik menunjukkan bahwa stored procedures memberikan peningkatan yang konsisten pada semua pengujian, terutama untuk query yang lebih kompleks, di mana pemrosesan data dapat dilakukan langsung di sisi server.




Gambar 7. Grafik performa terhadap penggunaan CPU

Grafik penggunaan CPU memperlihatkan perbedaan dampak dari masing-masing teknik optimasi. Pada teknik indexing, grafik menunjukkan penurunan penggunaan CPU dari rata-rata 9,4% menjadi hanya 3%. Penurunan ini disebabkan oleh efisiensi indeks dalam mengurangi kebutuhan proses komputasi saat mencari data, sehingga beban pada server menjadi lebih ringan. Grafik ini mengindikasikan bahwa indexing sangat cocok untuk query dengan volume data besar namun sederhana.

Sebaliknya, grafik penggunaan CPU pada teknik stored procedures menunjukkan peningkatan rata-rata dari 6% menjadi 9,4%. Meskipun terlihat ada peningkatan beban CPU, hal ini diimbangi dengan pengurangan durasi eksekusi query yang signifikan. Peningkatan penggunaan CPU ini terjadi karena stored procedures memindahkan sebagian besar pengolahan data dari aplikasi ke sisi server database, sehingga memberikan hasil yang lebih cepat.

Secara keseluruhan grafik durasi eksekusi dan penggunaan CPU menunjukkan bahwa indexing lebih efektif dalam mengurangi beban server secara keseluruhan, terutama pada query sederhana yang sering dilakukan, sedangkan stored procedures lebih optimal untuk mengolah query yang kompleks dengan durasi yang jauh lebih singkat, meskipun memerlukan peningkatan sumber daya CPU. Gabungan kedua teknik ini menghasilkan optimasi yang maksimal, di mana durasi eksekusi berkurang secara drastis tanpa mengorbankan performa server secara keseluruhan.

## 5. KESIMPULAN DAN SARAN

Kesimpulan berdasarkan penelitian ini, maka didapatkan bahwa implementasi metode indexing dan stored procedures dapat mengoptimalkan waktu eksekusi kueri memberikan peningkatan tanggung jawab dan efisiensi sistem secara keseluruhan, mendukung pengelolaan data akademik yang lebih baik bagi mahasiswa, dosen, dan pihak administrasi.optimasi kueri ini merupakan langkah strategi untuk menjawab tantangan pengelolaan data dalam lingkungan akademik yang terus

berkembang.Teknik pengindeksan secara efektif mempercepat proses pencarian data dengan menurunkan beban penggunaan CPU hingga rata-rata 3% dan mengurangi durasi pencarian hingga 93%. Sementara itu, penggunaan prosedur tersimpan, meskipun meningkatkan penggunaan CPU, memberikan durasi eksekusi yang jauh lebih optimal dengan peningkatan kinerja hingga 95%.

Saran untuk penelitian selanjutnya bisa mencoba menggunakan teknologi baru untuk menganalisis dan mengoptimalkan query secara otomatis. Selain itu, teknik ini bisa diuji pada platform lain untuk melihat apakah hasilnya tetap efektif di sistem yang berbeda. Selanjutnya, menggunakan dataset yang lebih besar juga penting untuk mengetahui apakah teknik ini tetap efisien saat menangani data yang lebih kompleks. Penelitian selanjutnya juga bisa mencoba menggabungkan beberapa teknik optimasi untuk mendapatkan hasil yang lebih maksimal.

## DAFTAR PUSTAKA


[1] Fadhol, S. (2021). Pengertian dan Manfaat Sistem Informasi Akademik Bagi Perguruan Tinggi & Mahasiswa | SEVIMA. SEVIMA. https://sevima.com/manfaat-sistem-informasi-akademik-bagi-perguruan-tinggi-mahasiswa/

[2] Kurnia Safitri, P., Wahyu Winarno, W., & Pramono, E. (2018). Optimasi Query untuk Sistem Informasi Penjadwalan Mata Pelajaran Sekolah Menggunakan View (Studi Kasus: SMK VIP Purworejo).Respati, 13(2). https://doi.org/10.35842/jtir.v13i2.259

[3] Himawan, H. (2018). Penggunaan Query Optimization Pada Sistem Informasi Akademik Untuk Penjadwalan Mata Kuliah Dan Pengisian Krs (Kartu Rencana Studi) Pada Universitas Matana. Faktor Exacta, 11(2), 104. https://doi.org/10.30998/faktorexacta.v11i2.2513

[4] Ramsi, A., Alfian, A., & Mukaromah, S. (2023). PERBANDINGAN PERFORMA WAKTU EKSEKUSI KUERI ANTARA DATABASE ORACLE DAN SQL SERVER. Prosiding Seminar Nasional Teknologi Dan Sistem Informasi, 3(1), 254–259. https://doi.org/10.33005/sitasi.v3i1.369

[5] 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.

[6] 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

[7] Siswanto, Heri, et al. "Optimasi Query Pada Human Resource Information System (HRIS) Di Universitas XYZ." Mercubuana-Yogya.ac.id, 2025, jmai.mercubuana-yogya.ac.id/index.php/jmai/article/view/53/13. Accessed 6 Jan. 2025.

[8] Iskandar, D., Fachruroji, M., Adi Septyo Wibowo, W., Khaerani, A. A., Teknologi Informasi, F., & Budi Luhur, U. (2022). *Database Tuning in Hospital Applications Using Table Indexing and Query Optimization*.

[9] Eirlangga, Y. S., Juledi, A. P., Manurung, K. H., Thoriq, M., & Syaputra, A. E. (2024). Dasar-dasar Sistem Basis Data. Rantauprapat: PT. Jasa Niaga Digital Indonesia. ISBN: 978-623-10-0276-1.

[10] Virmansyah, Y., Yulyanto Suladi, R., Bambang Lesmana, A., Teknologi Informasi, F., & Budi Luhur, U. (2022). Optimasi Database dengan Metode Index dan Partisi Tabel Database Postgresql pada Aplikasi E-Commerce. Studi pada Aplikasi Tokopintar.

[11] Affandi, R. (2024). Optimalisasi Algoritma Pencarian dalam Basis Data untuk Efisiensi Query. 1–8.

[12] Saputra, H. (2021). Pengembangan Sistem Informasi Akademik Pondok Pesantren Sumber Barokah Karawang untuk Meningkatan Kecepatan Modul Administrasi Keuangan. Tesis. Universitas Budi Luhur.

[13] Deska, A., Akbar, I., & Samidi, -. (2023). Tuning Database Pada Sistem Penerimaan Mahasiswa Baru Menggunakan Optimasi Query dan Indexing. Techno.Com, 22(1), 68–77. https://doi.org/10.33633/tc.v22i1.7047