

OPTIMALISASI KINERJA QUERY DAN BACKEND SISTEM PELAPORAN PENUNJANG MEDIS PADA SISTEM INFORMASI MANAJEMEN RUMAH SAKIT

Tutuko Rakha Hadyan, Sawali Wahyu
Teknik Informatika, Universitas Esa Unggul
Jakarta Barat, Indonesia
tutuko.rakha@student.esaunggul.ac.id

ABSTRAK

Sistem pelaporan penunjang medis pada Sistem Informasi Manajemen Rumah Sakit (SIMRS) berperan penting dalam mendukung proses operasional dan pengambilan keputusan. Namun, pada implementasi sistem yang berjalan, proses pelaporan sering mengalami waktu respons yang sangat lama akibat eksekusi query kompleks secara langsung pada basis data operasional. Pada kondisi awal, modul Laboratorium dengan volume data sekitar 100.000 baris dan modul Radiologi dengan sekitar 5.000 baris data memerlukan waktu pemrosesan lebih dari 20 menit, sedangkan modul Patologi Anatomi dengan sekitar 200 baris data membutuhkan waktu sekitar 6 detik. Penelitian ini bertujuan mengoptimalkan kinerja sistem pelaporan penunjang medis melalui penerapan strategi optimalisasi pada level basis data dan backend. Metode yang digunakan meliputi penerapan indexing dan materialized view untuk meningkatkan efisiensi query, pemrosesan asinkron pada backend, serta integrasi Redis dan MinIO sebagai mekanisme caching, pelacakan progres pembuatan laporan, dan penyimpanan hasil laporan dalam format JSON. Penelitian ini bertujuan mengoptimalkan kinerja sistem pelaporan penunjang medis melalui penerapan strategi optimalisasi pada level basis data dan backend. Metode yang digunakan meliputi penerapan indexing dan materialized view untuk meningkatkan efisiensi query, pemrosesan asinkron pada backend, serta integrasi Redis dan MinIO sebagai mekanisme caching, pelacakan progres pembuatan laporan, dan penyimpanan hasil laporan dalam format JSON.

Kata kunci : *Sistem Informasi Manajemen Rumah Sakit, Sistem Pelaporan, Optimasi Kinerja, Materialized View, Redis, MinIO*

1. PENDAHULUAN

Sistem Informasi Manajemen Rumah Sakit (SIMRS) merupakan komponen utama dalam pengelolaan data operasional rumah sakit yang mencakup layanan klinis, administrasi, penunjang medis, hingga pelaporan manajerial. Modul penunjang medis seperti laboratorium, radiologi, dan patologi anatomi menghasilkan data dalam jumlah besar setiap hari, sehingga menuntut sistem pelaporan yang memiliki kinerja cepat, stabil, dan andal. Kualitas SIMRS sangat dipengaruhi oleh kemampuan sistem dalam mengelola dan menyajikan data operasional secara efisien, khususnya pada modul dengan volume data tinggi [1]. Permasalahan performa sistem pelaporan pada sistem informasi rumah sakit telah banyak dibahas dalam penelitian sebelumnya, terutama terkait dampaknya terhadap keterlambatan pengambilan keputusan dan beban sistem operasional [2].

Dalam sistem informasi berbasis web, waktu respons (*response time*) merupakan salah satu indikator utama untuk menilai kualitas dan kinerja sistem. Respons sistem yang lambat dapat mengganggu alur kerja pengguna dan menurunkan efektivitas sistem secara keseluruhan. Literatur mengenai *responsiveness* menyebutkan bahwa keterlambatan respons sistem dalam skala menit hingga puluhan menit menunjukkan adanya permasalahan serius pada arsitektur dan mekanisme pemrosesan data [3]. Dalam konteks SIMRS, permasalahan waktu respons menjadi semakin krusial

karena keterlambatan akses laporan dapat berdampak langsung pada pengambilan keputusan operasional dan administratif. Permasalahan performa pada sistem pelaporan umumnya disebabkan oleh eksekusi query SQL yang kompleks secara langsung pada basis data operasional. Query pelaporan sering melibatkan banyak operasi *join*, filter berdasarkan rentang waktu, serta agregasi data historis dalam jumlah besar. Penelitian terdahulu menunjukkan bahwa eksekusi query kompleks tanpa optimasi yang memadai dapat meningkatkan beban basis data, memperpanjang waktu eksekusi, dan menurunkan stabilitas sistem [4].

Berbagai pendekatan telah dikembangkan untuk mengatasi permasalahan tersebut. Teknik optimasi basis data seperti *indexing* terbukti mampu mempercepat proses pencarian dan penyaringan data dengan menghindari *full table scan* pada tabel berukuran besar [5]. Selain itu, penggunaan *materialized view* dilaporkan dapat mengurangi waktu eksekusi query secara signifikan dengan menyimpan hasil pengolahan data secara fisik sehingga query tidak perlu dijalankan ulang setiap kali laporan diminta [6]. Di luar optimasi query SQL, pemanfaatan teknologi *in-memory caching* juga menjadi solusi yang banyak digunakan untuk meningkatkan performa sistem. Redis sebagai *in-memory data store* mampu menurunkan latensi akses data dan meningkatkan responsivitas aplikasi melalui mekanisme caching berbasis memori [7]. Redis juga dapat dimanfaatkan untuk menyimpan status dan progres pemrosesan data secara real-time, sehingga pengguna tetap

memperoleh respons meskipun proses pengolahan data berjalan dalam waktu lama [8].

Untuk pengelolaan hasil laporan yang berukuran besar dan bersifat tidak terstruktur, penggunaan *object storage* menjadi pendekatan yang semakin relevan. MinIO terbukti sebagai solusi penyimpanan objek yang efisien dan *scalable* untuk data non-relasional seperti laporan dalam format JSON [9]. Kombinasi Redis sebagai mekanisme caching dan MinIO sebagai penyimpanan objek terdistribusi dilaporkan mampu meningkatkan efisiensi sistem, mengurangi beban basis data operasional, serta memperbaiki stabilitas layanan pada sistem dengan volume data besar [10]. Meskipun berbagai penelitian telah membahas optimasi query, caching, dan penyimpanan objek, sebagian besar penelitian tersebut masih mengkaji teknik-teknik tersebut secara terpisah. Implementasi pendekatan optimasi yang terintegrasi—menggabungkan *materialized view*, *indexing*, pemrosesan backend asinkron, Redis untuk pelacakan progres, serta MinIO untuk penyimpanan hasil laporan—dalam konteks sistem pelaporan penunjang medis pada SIMRS masih jarang dilaporkan secara komprehensif. Padahal, karakteristik sistem pelaporan rumah sakit yang berjalan pada lingkungan nyata memiliki kompleksitas dan beban kerja yang berbeda dibandingkan sistem eksperimental.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengoptimalkan kinerja sistem pelaporan penunjang medis pada Sistem Informasi Manajemen Rumah Sakit melalui pendekatan terintegrasi pada level basis data dan backend. Optimalisasi dilakukan dengan menerapkan *indexing* dan *materialized view* pada basis data, mekanisme *refresh* berkala menggunakan *cronjob*, pemrosesan backend secara asinkron, serta integrasi Redis dan MinIO sebagai komponen utama *caching* dan penyimpanan hasil laporan. Melalui pengujian performa berbasis waktu respons dan stabilitas sistem, penelitian ini diharapkan dapat memberikan solusi aplikatif dan teruji untuk meningkatkan kinerja sistem pelaporan penunjang medis di lingkungan rumah sakit.

2. TINJAUAN PUSTAKA

Penelitian mengenai optimalisasi kinerja sistem basis data telah banyak dilakukan, khususnya pada sistem yang memproses data dalam skala besar. Azmat meneliti strategi *indexing* dalam SQL untuk meningkatkan efisiensi eksekusi query pada sistem basis data. Hasil penelitian tersebut menunjukkan bahwa penerapan *indexing* dan restrukturisasi query mampu menurunkan waktu eksekusi hingga 60% pada sistem dengan volume data tinggi. Penelitian lain yang dilakukan oleh Abbasi et al. (2024) mengkaji pendekatan *indexing* modern untuk komputasi paralel dan akselerasi pemrosesan query. Hasil penelitian tersebut menunjukkan bahwa penerapan teknik *parallel indexing* dapat meningkatkan efisiensi eksekusi query pada sistem basis data berskala besar.

Penelitian mengenai optimalisasi kinerja sistem basis data telah banyak dilakukan, khususnya pada sistem yang memproses data dalam skala besar. Azmat meneliti strategi *indexing* dalam SQL untuk meningkatkan efisiensi eksekusi query pada sistem basis data. Hasil penelitian tersebut menunjukkan bahwa penerapan *indexing* dan restrukturisasi query mampu menurunkan waktu eksekusi hingga 60% pada sistem dengan volume data tinggi. Penelitian lain yang dilakukan oleh Abbasi mengkaji pendekatan *indexing* modern untuk komputasi paralel dan akselerasi pemrosesan query. Hasil penelitian tersebut menunjukkan bahwa penerapan teknik *parallel indexing* dapat meningkatkan efisiensi eksekusi query pada sistem basis data berskala besar.

Meskipun berbagai penelitian tersebut telah membahas optimasi kinerja sistem melalui pendekatan *indexing*, manajemen penyimpanan data, maupun arsitektur sistem berbasis layanan, sebagian besar penelitian masih mengkaji teknik-teknik tersebut secara terpisah. Oleh karena itu, penelitian ini mengusulkan pendekatan optimalisasi yang terintegrasi dengan menggabungkan penggunaan *materialized view*, *indexing*, pemrosesan backend secara asinkron, serta integrasi Redis dan MinIO dalam satu arsitektur sistem pelaporan penunjang medis pada Sistem Informasi Manajemen Rumah Sakit.

Berdasarkan dua belas penelitian terdahulu yang disajikan pada tabel 1 sebelumnya, terdapat beberapa penelitian yang menjadi acuan utama dalam pengembangan optimalisasi pelaporan penunjang medis pada sistem SIMRS. Penelitian oleh Abbasi menunjukkan bahwa optimasi query SQL melalui *indexing* dan restrukturisasi dapat menurunkan waktu eksekusi secara signifikan pada sistem basis data berskala besar. Sementara itu, penelitian yang dilakukan Zulfa menegaskan bahwa Redis sangat efektif dalam meningkatkan performa sistem melalui mekanisme *caching* berbasis memori.

Meskipun penelitian-penelitian tersebut memberikan kontribusi penting dalam aspek *indexing*, *caching*, maupun penyimpanan objek, seluruh penelitian tersebut masih membahas teknik optimasi tersebut secara terpisah. Belum ada penelitian terdahulu yang mengintegrasikan beberapa teknik sekaligus yakni *Materialized View*, *cronjob*-based *refresh*, Redis *caching* untuk progres eksekusi, dan penyimpanan objek JSON melalui MinIO ke dalam satu arsitektur pelaporan yang utuh dan dapat diuji performanya secara langsung. Penelitian ini berbeda karena berfokus pada optimalisasi sistem pelaporan yang sudah berjalan pada lingkungan SIMRS nyata, bukan pengembangan sistem baru. Selain itu, penelitian ini menyajikan pendekatan terintegrasi yang mencakup optimasi query, pemrosesan paralel, *caching* real-time, hingga penyimpanan objek, sehingga mampu memberikan solusi holistik terhadap permasalahan keterlambatan pelaporan penunjang

medis yang belum terselesaikan oleh penelitian sebelumnya.

2.1. Sistem Pelaporan pada Sistem Informasi Manajemen Rumah Sakit

Sistem pelaporan merupakan salah satu komponen penting dalam Sistem Informasi Manajemen Rumah Sakit (SIMRS) yang berfungsi menyajikan informasi operasional dan klinis untuk mendukung proses pengambilan keputusan manajerial maupun pelayanan medis. Informasi yang dihasilkan dari sistem pelaporan digunakan untuk evaluasi kinerja layanan, perencanaan sumber daya, serta pemantauan aktivitas operasional rumah sakit secara berkala [1].

Seiring dengan meningkatnya volume data transaksi penunjang medis, sistem pelaporan pada SIMRS sering menghadapi permasalahan kinerja, khususnya terkait waktu respons dan stabilitas layanan. Proses pelaporan umumnya melibatkan query kompleks dengan operasi *JOIN*, agregasi data, dan penyaringan rentang waktu, yang apabila dijalankan langsung pada basis data operasional dapat menyebabkan keterlambatan respons dan gangguan pada layanan sistem lainnya.

2.2. Optimasi Query Basis Data untuk Sistem Pelaporan

Optimasi query merupakan pendekatan yang umum digunakan untuk meningkatkan kinerja sistem basis data, terutama pada sistem pelaporan yang bersifat *read-intensive* dan memproses volume data besar. Tanpa mekanisme optimasi yang tepat, eksekusi query kompleks dapat menyebabkan *full table scan* dan meningkatkan beban kerja basis data secara signifikan [6].

Salah satu teknik optimasi yang banyak diterapkan adalah penggunaan *indexing* pada kolom yang sering digunakan dalam operasi *JOIN* dan *filtering*. Penerapan indeks memungkinkan sistem basis data mempercepat proses pencarian data dan mengurangi waktu eksekusi query. Selain itu, *materialized view* digunakan untuk menyimpan hasil eksekusi query secara fisik, sehingga sistem pelaporan dapat mengakses data yang telah diproses tanpa harus menjalankan ulang query kompleks setiap kali laporan diminta.

2.3. Pemrosesan Asinkron dan Paralel Backend

Pemrosesan sinkron pada sistem backend mengharuskan aplikasi menunggu seluruh proses selesai sebelum memberikan respons kepada pengguna. Pendekatan ini berpotensi menimbulkan masalah pada aplikasi dengan beban pemrosesan tinggi, karena waktu respons menjadi sangat bergantung pada durasi proses yang dijalankan [7].

Sebagai alternatif, pemrosesan asinkron dan paralel memungkinkan sistem backend untuk tetap responsif dengan menjalankan proses berat di latar belakang. Pembagian proses ke dalam beberapa tugas kecil yang dijalankan secara paralel menggunakan

thread pool dapat menurunkan waktu pemrosesan total dan meningkatkan skalabilitas sistem. Pendekatan ini banyak diterapkan pada aplikasi berbasis layanan web yang membutuhkan kinerja tinggi dan stabilitas layanan.

2.4. Caching dan Object Storage dalam Sistem Pelaporan

Caching merupakan teknik yang digunakan untuk menyimpan data sementara pada media dengan kecepatan akses tinggi guna mengurangi frekuensi akses ke basis data utama. *In-memory data store* seperti Redis banyak digunakan untuk kebutuhan caching karena mampu memberikan akses data dengan latensi rendah serta mendukung pengelolaan status dan informasi sementara pada sistem backend [8].

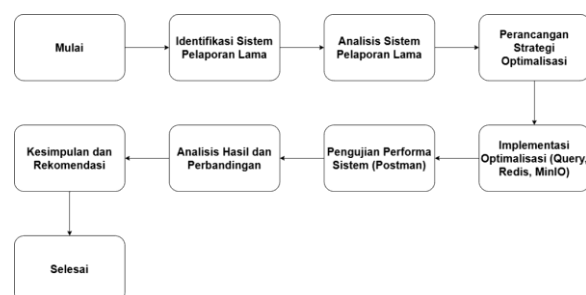
Selain caching, penyimpanan hasil pemrosesan laporan dalam bentuk file juga menjadi pendekatan yang efektif untuk meningkatkan efisiensi sistem pelaporan. *Object storage* seperti MinIO memungkinkan penyimpanan data tidak terstruktur, seperti file JSON hasil laporan, dengan skalabilitas tinggi dan akses yang efisien. Dengan memanfaatkan *object storage*, sistem dapat melayani permintaan laporan yang sama tanpa perlu menjalankan ulang proses pengolahan data.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan melakukan optimalisasi langsung pada sistem pelaporan penunjang medis yang telah berjalan pada lingkungan Sistem Informasi Manajemen Rumah Sakit (SIMRS). Fokus penelitian diarahkan pada peningkatan kinerja sistem, khususnya waktu respons dan stabilitas layanan, melalui penerapan optimasi pada level basis data dan arsitektur backend. Metodologi penelitian disusun untuk memastikan bahwa solusi yang diusulkan bersifat aplikatif, terukur, dan dapat dievaluasi secara objektif.

3.1. Tahapan Penelitian Optimalisasi Sistem

Metodologi optimalisasi sistem dalam penelitian ini dilakukan melalui beberapa tahapan utama sebagai berikut.



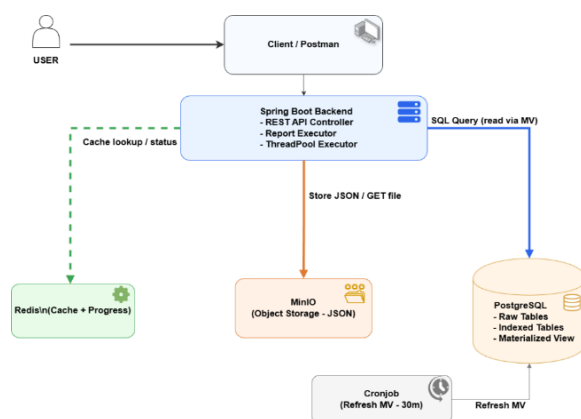
Gambar 1. Alur Optimasi Sistem Pelaporan Penunjang Medis

Tahapan optimalisasi sistem pelaporan penunjang medis pada penelitian ini ditunjukkan pada Gambar 1. Proses penelitian diawali dengan

identifikasi dan analisis sistem pelaporan yang berjalan untuk menemukan permasalahan kinerja. Selanjutnya dilakukan perancangan strategi optimalisasi yang mencakup optimasi query dan arsitektur backend. Tahap implementasi meliputi penerapan optimalisasi pada basis data dan backend menggunakan Redis dan MinIO. Pengujian performa sistem dilakukan untuk mengukur waktu respons dan stabilitas layanan, yang kemudian dianalisis sebagai dasar penarikan kesimpulan dan rekomendasi.

3.2. Rancangan Arsitektur Optimalisasi Basis Data

Rancangan arsitektur optimalisasi basis data disusun sebagai kerangka konseptual untuk memastikan pengelolaan data yang efisien, terstruktur, dan mampu mendukung kinerja sistem secara optimal. Arsitektur ini dirancang dengan mempertimbangkan kebutuhan pemrosesan data, skalabilitas sistem, serta integrasi antar komponen, sehingga mampu meningkatkan kecepatan akses, keandalan, dan konsistensi data dalam mendukung fungsi utama aplikasi.



Gambar 2. Diagram Arsitektur Optimalisasi

Pada Gambar 2. Optimalisasi pada level basis data dilakukan dengan menerapkan *indexing* dan *materialized view*. Index diterapkan pada kolom-kolom yang sering digunakan dalam operasi *join* dan *filter* untuk mempercepat proses pencarian data dan menghindari *full table scan* [11]. Selain itu, *materialized view* digunakan untuk menyimpan hasil eksekusi query kompleks secara fisik, sehingga proses pelaporan tidak lagi bergantung langsung pada tabel transaksi utama [12]. *Materialized view* diperbarui secara berkala menggunakan *cronjob* setiap 30 menit agar data tetap mendekati kondisi real-time tanpa membebani sistem operasional [6].

Pada sisi backend, sistem pelaporan diubah dari pemrosesan sinkron menjadi pemrosesan asinkron. Permintaan laporan diproses secara bertahap dengan membagi data ke dalam beberapa segmen menggunakan mekanisme *pagination* [13]. Setiap segmen diproses secara paralel menggunakan *thread pool executor*, sehingga waktu pemrosesan total dapat ditekan tanpa meningkatkan beban sistem secara

berlebihan. Redis diintegrasikan sebagai *in-memory data store* untuk mencatat status dan progres pembuatan laporan secara real-time [14]. Redis juga digunakan untuk mencegah eksekusi laporan yang sama secara berulang ketika permintaan dengan parameter identik diajukan kembali [7].

Hasil laporan yang telah selesai diproses disimpan dalam format JSON pada *object storage* MinIO. Pendekatan ini memungkinkan laporan yang telah dihasilkan digunakan kembali tanpa perlu menjalankan proses query dan pengolahan ulang. Penggunaan MinIO sebagai media penyimpanan objek dipilih karena kemampuannya dalam menyediakan akses cepat dan *scalable* untuk data non-relasional [9]. Integrasi Redis dan MinIO diharapkan mampu mengurangi beban basis data dan meningkatkan efisiensi sistem pelaporan secara keseluruhan [10].

3.3. Metode Pengujian dan Evaluasi

Evaluasi kinerja sistem dilakukan dengan membandingkan kondisi sebelum dan sesudah optimalisasi. Parameter utama yang digunakan adalah waktu respons sistem, yang diukur menggunakan Postman sebagai alat pengujian REST API [15]. Pengujian dilakukan dengan menjalankan permintaan laporan pada setiap modul penunjang medis sebanyak sepuluh kali percobaan dengan parameter dan rentang waktu yang sama.

Selain waktu respons, stabilitas sistem juga dievaluasi berdasarkan keberadaan kegagalan respons seperti *Gateway Timeout* (HTTP 504). Hasil pengujian dicatat dan dianalisis untuk mengukur tingkat peningkatan kinerja sistem secara kuantitatif. Dengan metode ini, efektivitas optimalisasi dapat diukur secara objektif dan dapat dibandingkan dengan hasil penelitian terdahulu.

3.4. Analisis Stabilitas Sistem

Selain peningkatan waktu respons, stabilitas sistem menjadi aspek penting dalam evaluasi kinerja sistem pelaporan penunjang medis. Stabilitas sistem pada penelitian ini dianalisis berdasarkan kemampuan sistem dalam menangani permintaan laporan tanpa mengalami kegagalan respons, khususnya kesalahan *Gateway Timeout* (HTTP 504), serta konsistensi sistem dalam merespons permintaan berulang dengan parameter yang sama.

Pada kondisi sebelum optimalisasi, sistem pelaporan sering mengalami kegagalan respons ketika permintaan laporan diajukan dengan rentang waktu data yang panjang dan volume data besar. Proses eksekusi query dan pengolahan data dilakukan secara sinkron, sehingga backend harus menunggu seluruh proses selesai sebelum memberikan respons kepada pengguna. Kondisi ini menyebabkan server melewati batas waktu respons yang diizinkan, yang ditandai dengan munculnya kesalahan HTTP 504 pada beberapa percobaan pengujian.

Setelah optimalisasi diterapkan, hasil pengujian menunjukkan bahwa kegagalan respons tersebut tidak

lagi ditemukan pada seluruh modul yang diuji. Sistem mampu memberikan respons awal secara konsisten meskipun proses pembuatan laporan masih berjalan di latar belakang. Perubahan ini menunjukkan peningkatan stabilitas sistem yang signifikan, terutama pada skenario permintaan laporan dengan volume data besar.

Peningkatan stabilitas ini dipengaruhi oleh penerapan pemrosesan asinkron pada backend serta integrasi Redis sebagai mekanisme pencatatan status dan progres pembuatan laporan. Dengan memanfaatkan Redis, sistem dapat mencatat status proses secara real-time dan mencegah eksekusi laporan yang sama secara bersamaan. Selain itu, pemisahan proses pengambilan data dan pengembalian respons memungkinkan backend tetap responsif tanpa harus menunggu seluruh proses pengolahan data selesai.

Hasil analisis stabilitas sistem menunjukkan bahwa optimalisasi yang diterapkan tidak hanya meningkatkan kecepatan sistem, tetapi juga meningkatkan keandalan layanan pelaporan secara keseluruhan. Sistem menjadi lebih toleran terhadap beban data besar dan mampu menangani permintaan laporan secara berulang tanpa mengalami kegagalan respons, sehingga lebih sesuai untuk digunakan pada lingkungan operasional rumah sakit yang membutuhkan layanan sistem yang stabil dan berkelanjutan.

3.5. Pembahasan Hasil

Hasil pengujian yang telah disajikan pada Subbab 3.1 hingga 3.3 menunjukkan bahwa strategi optimalisasi yang diterapkan pada sistem pelaporan penunjang medis memberikan dampak yang signifikan terhadap kinerja dan stabilitas sistem. Peningkatan waktu respons hingga lebih dari 99% pada modul Laboratorium dan Radiologi menegaskan bahwa permasalahan utama sistem sebelumnya bukan terletak pada infrastruktur jaringan atau perangkat keras, melainkan pada mekanisme pengolahan data dan arsitektur pemrosesan sistem.

Penerapan *indexing* dan *materialized view* pada basis data terbukti efektif dalam mengurangi beban eksekusi query kompleks yang sebelumnya dijalankan langsung pada tabel transaksi operasional. Temuan ini sejalan dengan penelitian Albert dan Blas serta Ramdoss, yang menyatakan bahwa penggunaan struktur data terindeks dan *materialized view* dapat mempercepat proses pengambilan data dan mengurangi waktu eksekusi query pada sistem dengan volume data besar [6], [11]. Dalam konteks penelitian ini, penggunaan *materialized view* juga berperan penting dalam menjaga performa basis data operasional agar tidak terganggu oleh permintaan laporan yang bersifat berat.

Dari sisi backend, perubahan mekanisme pemrosesan dari sinkron menjadi asinkron memberikan kontribusi besar terhadap peningkatan responsivitas dan stabilitas sistem. Sistem tidak lagi

menunggu seluruh proses pengambilan dan pengolahan data selesai sebelum memberikan respons kepada pengguna. Pendekatan ini mendukung temuan Joshi dan Zulfa et al., yang menyatakan bahwa pemrosesan asinkron dan pemanfaatan *in-memory caching* dapat meningkatkan keandalan sistem dan mengurangi risiko kegagalan respons pada aplikasi berbasis web [7], [8].

Integrasi Redis sebagai mekanisme pencatatan status dan progres pembuatan laporan memungkinkan sistem memberikan umpan balik secara real-time kepada pengguna. Selain meningkatkan transparansi proses, mekanisme ini juga mencegah eksekusi laporan yang sama secara bersamaan, sehingga menghindari pemborosan sumber daya sistem. Hasil penelitian ini memperkuat pandangan bahwa Redis tidak hanya efektif sebagai media caching data, tetapi juga sebagai komponen pendukung pengelolaan proses backend pada sistem berskala besar.

Pemanfaatan MinIO sebagai media penyimpanan hasil laporan dalam format JSON turut berkontribusi dalam meningkatkan efisiensi sistem pelaporan. Dengan menyimpan hasil laporan yang telah selesai diproses, sistem dapat melayani permintaan laporan yang sama tanpa perlu menjalankan ulang proses pengolahan data. Pendekatan ini sejalan dengan penelitian Malhotra dan Saqib serta Zhang et al., yang menunjukkan bahwa *object storage* efektif digunakan untuk menyimpan data tidak terstruktur dan mendukung sistem dengan kebutuhan akses data berulang [9], [10].

Secara keseluruhan, pembahasan hasil menunjukkan bahwa optimalisasi sistem pelaporan penunjang medis melalui pendekatan terintegrasi pada level basis data dan backend mampu menjawab permasalahan performa dan stabilitas yang sebelumnya terjadi. Pendekatan ini tidak hanya relevan secara teknis, tetapi juga aplikatif untuk diterapkan pada sistem informasi rumah sakit yang memiliki karakteristik data besar dan kebutuhan layanan yang bersifat kritis.

4. HASIL DAN PEMBAHASAN

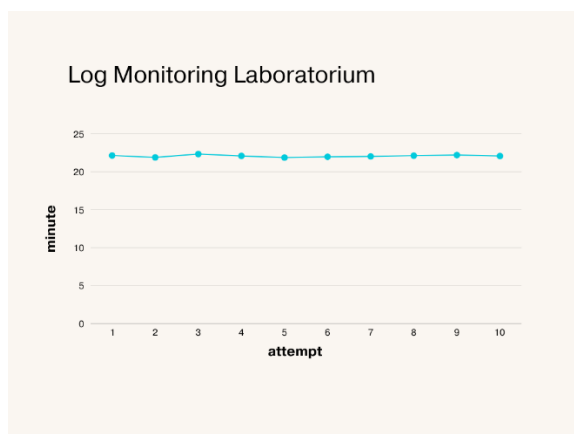
4.1. Hasil Pengujian Sebelum Optimasi

Sebelum penerapan optimalisasi, sistem pelaporan penunjang medis pada Sistem Informasi Manajemen Rumah Sakit (SIMRS) memproses setiap permintaan laporan secara **sinkron**. Ketika pengguna mengajukan permintaan laporan, backend secara langsung mengeksekusi query kompleks yang melibatkan banyak operasi *JOIN*, filter rentang waktu, serta pengolahan data pada tabel transaksi operasional. Seluruh proses pengambilan dan pengolahan data tersebut harus diselesaikan terlebih dahulu sebelum sistem dapat memberikan respons kepada pengguna.

Arsitektur pemrosesan sinkron ini menyebabkan backend berada dalam kondisi *blocking* selama proses eksekusi query berlangsung. Akibatnya, waktu respons sistem sangat bergantung pada durasi eksekusi query dan volume data yang diproses. Pada modul

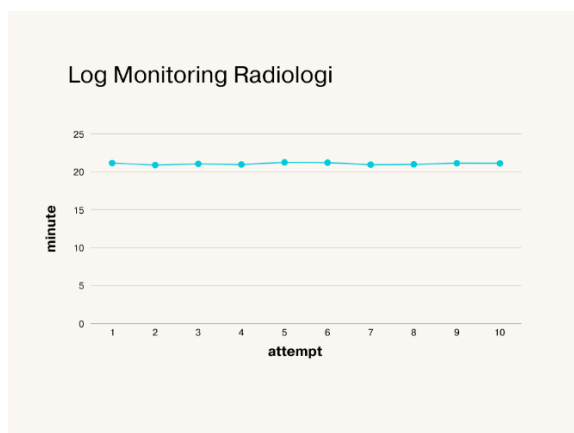
dengan volume data besar seperti Laboratorium dan Radiologi, kondisi ini berdampak pada waktu tunggu yang sangat lama dan berpotensi menimbulkan kegagalan respons ketika durasi pemrosesan melebihi batas waktu layanan.

Untuk memperoleh gambaran kinerja sistem sebelum optimalisasi, dilakukan pengujian waktu respons sistem menggunakan alat pengujian REST API. Pengujian dilakukan pada tiga modul penunjang medis, yaitu Laboratorium, Radiologi, dan Patologi Anatomi. Setiap pengujian dilakukan dengan parameter dan rentang waktu data yang sama, serta diulang sebanyak sepuluh kali percobaan untuk memastikan konsistensi hasil pengukuran.



Gambar 3. Grafik Waktu Respons Modul Laboratorium Sebelum Optimalisasi

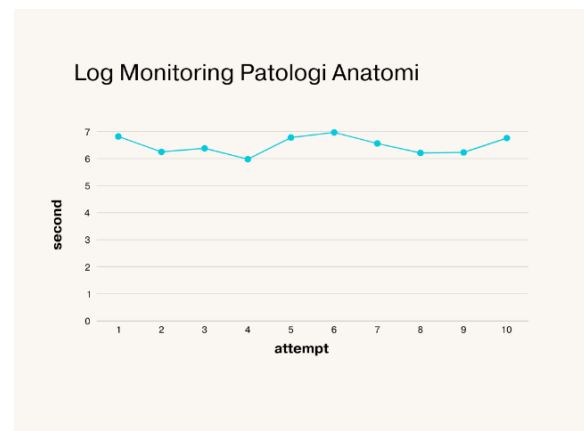
Hasil pengujian pada modul Laboratorium menunjukkan bahwa waktu respons sistem sebelum optimalisasi berada pada kisaran 20 menit pada seluruh percobaan. Nilai waktu respons yang sangat tinggi ini mengindikasikan bahwa sistem harus memproses data dalam jumlah besar secara langsung dari tabel operasional tanpa mekanisme percepatan atau pemisahan beban pemrosesan.



Gambar 4. Grafik Waktu Respons Modul Radiologi Sebelum Optimalisasi

Pada modul Radiologi, hasil pengujian menunjukkan pola serupa dengan modul

Laboratorium. Seluruh percobaan menghasilkan waktu respons yang sangat panjang, mencapai lebih dari seribu detik. Hal ini disebabkan oleh kompleksitas query pelaporan radiologi yang melibatkan banyak tabel serta volume data historis yang besar.



Gambar 5. Grafik Waktu Respons Modul Radiologi Sebelum Optimalisasi

Berbeda dengan dua modul sebelumnya, modul Patologi Anatomi menunjukkan waktu respons yang relatif lebih rendah sebelum optimalisasi. Kondisi ini disebabkan oleh volume data yang lebih kecil dalam rentang waktu pengujian. Meskipun demikian, proses pelaporan tetap berjalan secara sinkron, sehingga backend tetap harus menunggu seluruh proses pengolahan data selesai sebelum memberikan respons.

Berdasarkan grafik hasil pengujian, dapat disimpulkan bahwa sistem pelaporan penunjang medis sebelum optimalisasi memiliki kinerja yang belum efisien, terutama pada modul dengan volume data besar. Waktu respons yang sangat panjang serta sifat pemrosesan sinkron menjadi permasalahan utama yang melatarbelakangi perlunya penerapan strategi optimalisasi pada level basis data dan arsitektur backend.

4.2. Hasil Pengujian Setelah Optimalisasi

Sebelum penerapan optimalisasi, sistem pelaporan penunjang medis pada SIMRS memproses setiap permintaan laporan secara sinkron. Ketika pengguna mengajukan permintaan laporan, backend secara langsung mengeksekusi query kompleks yang melibatkan banyak operasi JOIN, filter rentang waktu, dan pengolahan data pada tabel transaksi operasional. Seluruh proses pengambilan dan pengolahan data harus diselesaikan terlebih dahulu sebelum sistem dapat memberikan respons kepada pengguna. Mekanisme ini menyebabkan waktu respons yang sangat panjang, terutama pada modul dengan volume data besar seperti Laboratorium dan Radiologi.

Pengujian waktu respons sistem dilakukan untuk mengevaluasi dampak penerapan optimalisasi terhadap kinerja sistem pelaporan penunjang medis. Pengujian difokuskan pada tiga modul utama, yaitu Laboratorium, Radiologi, dan Patologi Anatomi, yang

memiliki karakteristik volume data berbeda. Setiap pengujian dilakukan dengan parameter dan rentang waktu data yang sama, serta diulang sebanyak sepuluh kali percobaan untuk memastikan konsistensi hasil.

Parameter utama yang diukur pada pengujian ini adalah waktu respons sistem, yaitu selang waktu sejak permintaan laporan dikirimkan hingga sistem memberikan respons awal kepada pengguna. Pada kondisi sebelum optimalisasi, waktu respons mencerminkan keseluruhan durasi eksekusi query dan pemrosesan data secara sinkron. Setelah optimalisasi, waktu respons merepresentasikan kemampuan sistem dalam merespons permintaan laporan dengan memanfaatkan mekanisme pemrosesan asinkron dan *caching*. Pengukuran waktu respons digunakan sebagai indikator utama kinerja sistem karena secara langsung merepresentasikan efisiensi pemrosesan backend dan basis data sebagaimana digunakan pada penelitian optimasi sistem basis data dan aplikasi berbasis Java [16].

Tabel 2. Tabel Perbandingan Respon Time Laboratorium

No	Respon Time Sebelum Optimasi (detik)	Respon Time Setelah Optimasi (detik)	Improve (%)
1	1329.32	6.28	99.53%
2	1314.54	8.75	99.33%
3	1340.08	9.63	99.28%
4	1325.46	7.01	99.47%
5	1312.31	7.87	99.40%
6	1318.54	5.88	99.55%
7	1321.96	5.95	99.55%
8	1327.77	6.45	99.51%
9	1332.83	6.34	99.52%
10	1324.28	6.69	99.49%

Tabel 3. Tabel Perbandingan Respon Time Radiologi

No	Respon Time Sebelum Optimasi (detik)	Respon Time Setelah Optimasi (detik)	Improve (%)
1	1269.26	0.56	99.96%
2	1253.16	0.44	99.96%
3	1263.08	0.54	99.96%
4	1257.12	0.42	99.97%
5	1275.02	0.53	99.96%
6	1272.31	0.45	99.96%
7	1256.08	0.46	99.96%
8	1258.11	0.42	99.97%
9	1268.05	0.43	99.97%
10	1267.05	0.42	99.97%

Hasil pengujian menunjukkan adanya penurunan waktu respons yang sangat signifikan pada seluruh modul yang diuji. Pada modul Laboratorium, waktu respons sistem setelah optimalisasi berada pada kisaran kurang lebih 5 sampai 10 detik, dibandingkan dengan waktu respons sebelumnya yang mencapai lebih dari seribu detik. Hasil detail pengujian pada modul Laboratorium disajikan pada Tabel 1.

Modul Radiologi juga menunjukkan hasil serupa, di mana waktu respons sistem setelah optimalisasi berada di bawah satu detik pada seluruh percobaan. Sebelum optimalisasi, proses pelaporan pada modul ini memerlukan waktu lebih dari seribu detik. Rincian hasil pengujian modul Radiologi ditampilkan pada Tabel 2.

Tabel 4. Tabel Perbandingan Respon Time Patologi Anatomi

No	Respon Time Sebelum Optimasi (detik)	Respon Time Setelah Optimasi (detik)	Improve (%)
1	6.82	0.26	96.14%
2	6.25	0.31	94.91%
3	6.38	0.16	97.37%
4	5.98	0.15	97.34%
5	6.78	0.14	97.80%
6	6.97	0.15	97.83%
7	6.56	0.13	97.88%
8	6.21	0.14	97.73%
9	6.23	0.14	97.66%
10	6.76	0.15	97.66%

Pada modul Patologi Anatomi, waktu respons sebelum optimalisasi relatif lebih kecil dibandingkan modul lainnya. Hal ini disebabkan oleh volume data yang lebih rendah dalam rentang waktu pengujian. Meskipun demikian, optimalisasi tetap memberikan peningkatan kinerja dengan menurunkan waktu respons menjadi kurang dari satu detik, tanpa menimbulkan degradasi performa sistem.

Hasil pengujian ini menunjukkan bahwa strategi optimalisasi yang diterapkan efektif dalam meningkatkan responsivitas sistem pelaporan, terutama pada modul dengan volume data besar. Selain itu, hasil yang konsisten pada beberapa kali pengujian menunjukkan bahwa peningkatan kinerja yang diperoleh bersifat stabil dan bukan hasil dari kondisi pengujian tertentu.

4.3. Analisis Peningkatan Kinerja Sistem

Subbab ini menyajikan analisis kuantitatif terhadap peningkatan kinerja sistem pelaporan penunjang medis setelah penerapan optimalisasi. Analisis dilakukan dengan membandingkan waktu respons sistem sebelum dan sesudah optimalisasi yang telah disajikan pada Subbab 3.1. Parameter utama yang dianalisis adalah persentase peningkatan waktu respons (*performance improvement*), yang digunakan untuk mengukur efektivitas strategi optimalisasi yang diterapkan.

Persentase peningkatan kinerja dihitung berdasarkan selisih waktu respons sebelum dan sesudah optimalisasi terhadap waktu respons awal. Pendekatan ini dipilih untuk memberikan gambaran yang lebih jelas mengenai tingkat efisiensi yang berhasil dicapai oleh sistem setelah optimalisasi, khususnya pada modul dengan volume data besar.

Tabel 5. Tabel Persentase Optimasi

No	Modul Penunjang	Peningkatan Kinerja (%)
1	Laboratorium	99,46%
2	Radiologi	99,96%
3	Patologi Anatomi	97,03%

Berdasarkan Tabel 4, modul Laboratorium dan Radiologi menunjukkan tingkat peningkatan kinerja yang sangat tinggi, yaitu lebih dari 99%. Peningkatan ini mengindikasikan bahwa strategi optimalisasi yang diterapkan sangat efektif dalam menangani beban data besar dan query kompleks yang sebelumnya menyebabkan keterlambatan signifikan. Perubahan arsitektur pemrosesan dari sinkron menjadi asinkron, serta pemanfaatan struktur data hasil pemrosesan, berkontribusi besar terhadap peningkatan ini.

Pada modul Patologi Anatomi, persentase peningkatan kinerja tercatat kurang lebih sebesar 97%. Meskipun nilai peningkatannya lebih rendah dibandingkan dua modul lainnya, hasil ini tetap menunjukkan adanya peningkatan performa yang signifikan. Perbedaan tersebut disebabkan oleh karakteristik volume data yang relatif kecil pada modul Patologi Anatomi, sehingga dampak optimalisasi terhadap waktu respons tidak sebesar pada modul dengan volume data tinggi.

Secara keseluruhan, hasil analisis menunjukkan bahwa optimalisasi sistem pelaporan penunjang medis berhasil meningkatkan kinerja sistem secara signifikan dan konsisten. Persentase peningkatan yang tinggi pada modul dengan beban data besar memperkuat temuan bahwa pendekatan optimasi terintegrasi pada level basis data dan backend efektif untuk meningkatkan efisiensi sistem pelaporan di lingkungan rumah sakit.

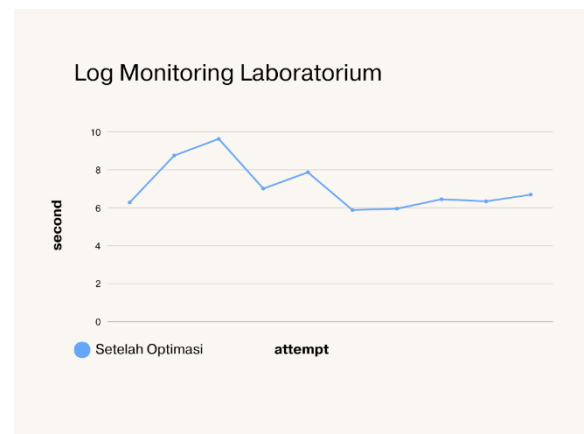
4.4. Analisis Perbandingan Kondisi dan Sesudah Optimasi

Analisis perbandingan dilakukan untuk mengevaluasi perubahan karakteristik kinerja sistem pelaporan penunjang medis sebelum dan sesudah penerapan optimalisasi. Perbandingan difokuskan pada aspek waktu respons, pola pemrosesan data, serta stabilitas sistem dalam menangani permintaan laporan secara berulang dengan parameter yang sama.

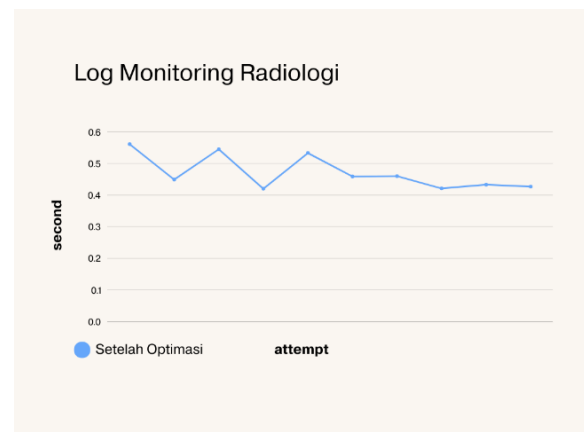
Pada kondisi sebelum optimalisasi, sistem pelaporan memproses permintaan laporan secara sinkron. Setiap permintaan laporan mengharuskan backend mengeksekusi query kompleks secara langsung pada tabel transaksi operasional dan menunggu seluruh proses pengambilan serta pengolahan data selesai sebelum memberikan respons kepada pengguna. Pola pemrosesan ini menyebabkan waktu respons yang sangat panjang, terutama pada modul dengan volume data besar seperti Laboratorium dan Radiologi, yang pada beberapa pengujian mencapai lebih dari seribu detik. Selain berdampak pada lamanya waktu tunggu pengguna, pendekatan sinkron ini juga meningkatkan risiko kegagalan

respons ketika beban pemrosesan melebihi batas waktu layanan.

Setelah optimalisasi diterapkan, mekanisme pemrosesan sistem mengalami perubahan yang signifikan. Sistem tidak lagi menunggu seluruh proses pengolahan data selesai sebelum memberikan respons. Backend memberikan respons awal secara cepat kepada pengguna, sementara proses pengambilan data dan pembentukan laporan dijalankan secara asinkron di latar belakang. Perubahan arsitektur pemrosesan ini menyebabkan waktu respons sistem turun secara drastis hingga berada pada kisaran detik, bahkan sub-detik pada modul tertentu.

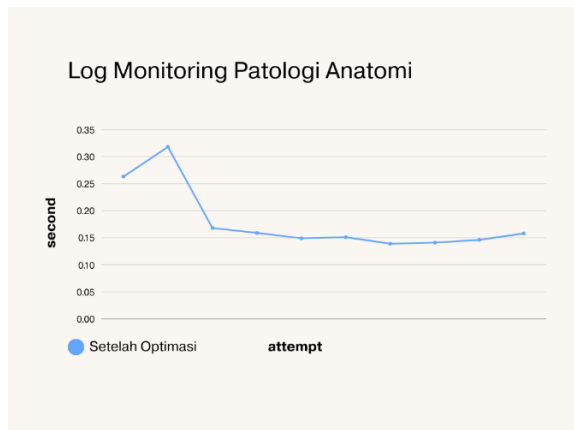


Gambar 6. Grafik Waktu Respons Modul Laboratorium Setelah Optimalisasi



Gambar 7. Grafik Waktu Respons Modul Radiologi Setelah Optimalisasi

Pada modul Patologi Anatomi, hasil pengujian setelah optimalisasi juga menunjukkan adanya fluktuasi waktu respons antar percobaan. Seluruh nilai waktu respons berada di bawah satu detik, dengan variasi yang relatif kecil. Hal ini menunjukkan bahwa penerapan optimalisasi tidak menurunkan kinerja pada modul dengan volume data relatif kecil, serta tidak menimbulkan overhead tambahan yang signifikan pada proses pelaporan.



Gambar 8. Grafik Waktu Respons Modul Patologi Anatomi Setelah Optimalisasi

Secara keseluruhan, perbandingan kondisi sebelum dan sesudah optimalisasi menunjukkan bahwa peningkatan kinerja sistem tidak hanya bersifat kuantitatif dalam bentuk penurunan waktu respons, tetapi juga bersifat struktural dalam cara sistem menangani permintaan laporan. Perubahan arsitektur pemrosesan dari sinkron menjadi asinkron menjadi faktor utama yang memungkinkan sistem pelaporan beroperasi secara lebih cepat, stabil, dan efisien pada berbagai karakteristik modul penunjang medis.

4.5. Analisis Stabilitas Sistem

Selain peningkatan waktu respons, stabilitas sistem menjadi aspek penting dalam evaluasi kinerja sistem pelaporan penunjang medis. Stabilitas sistem pada penelitian ini dianalisis berdasarkan kemampuan sistem dalam menangani permintaan laporan tanpa mengalami kegagalan respons, khususnya kesalahan *Gateway Timeout* (HTTP 504), serta konsistensi sistem dalam merespons permintaan berulang dengan parameter yang sama.

Pada kondisi sebelum optimalisasi, sistem pelaporan sering mengalami kegagalan respons ketika permintaan laporan diajukan dengan rentang waktu data yang panjang dan volume data besar. Proses eksekusi query dan pengolahan data dilakukan secara sinkron, sehingga backend harus menunggu seluruh proses selesai sebelum memberikan respons kepada pengguna. Kondisi ini menyebabkan server melewati batas waktu respons yang diizinkan, yang ditandai dengan munculnya kesalahan HTTP 504 pada beberapa percobaan pengujian.

Setelah optimalisasi diterapkan, hasil pengujian menunjukkan bahwa kegagalan respons tersebut tidak lagi ditemukan pada seluruh modul yang diuji. Sistem mampu memberikan respons awal secara konsisten meskipun proses pembuatan laporan masih berjalan di latar belakang. Perubahan ini menunjukkan peningkatan stabilitas sistem yang signifikan, terutama pada skenario permintaan laporan dengan volume data besar.

Peningkatan stabilitas ini dipengaruhi oleh penerapan pemrosesan asinkron pada backend serta

integrasi Redis sebagai mekanisme pencatatan status dan progres pembuatan laporan. Dengan memanfaatkan Redis, sistem dapat mencatat status proses secara real-time dan mencegah eksekusi laporan yang sama secara bersamaan. Selain itu, pemisahan proses pengambilan data dan pengembalian respons memungkinkan backend tetap responsif tanpa harus menunggu seluruh proses pengolahan data selesai.

Hasil analisis stabilitas sistem menunjukkan bahwa optimalisasi yang diterapkan tidak hanya meningkatkan kecepatan sistem, tetapi juga meningkatkan keandalan layanan pelaporan secara keseluruhan. Sistem menjadi lebih toleran terhadap beban data besar dan mampu menangani permintaan laporan secara berulang tanpa mengalami kegagalan respons, sehingga lebih sesuai untuk digunakan pada lingkungan operasional rumah sakit yang membutuhkan layanan sistem yang stabil dan berkelanjutan.

4.6. Pembahasan Hasil

Hasil pengujian yang telah disajikan pada Subbab 4.1 hingga 4.4 menunjukkan bahwa strategi optimalisasi yang diterapkan pada sistem pelaporan penunjang medis memberikan dampak yang signifikan terhadap kinerja dan stabilitas sistem. Peningkatan waktu respons hingga lebih dari 99% pada modul Laboratorium dan Radiologi menegaskan bahwa permasalahan utama sistem sebelumnya bukan terletak pada infrastruktur jaringan atau perangkat keras, melainkan pada mekanisme pengolahan data dan arsitektur pemrosesan sistem.

Penerapan *indexing* dan *materialized view* pada basis data terbukti efektif dalam mengurangi beban eksekusi query kompleks yang sebelumnya dijalankan langsung pada tabel transaksi operasional. Temuan ini sejalan dengan penelitian Albert dan Blas serta Ramdoss, yang menyatakan bahwa penggunaan struktur data terindeks dan *materialized view* dapat mempercepat proses pengambilan data dan mengurangi waktu eksekusi query pada sistem dengan volume data besar [6], [11]. Dalam konteks penelitian ini, penggunaan *materialized view* juga berperan penting dalam menjaga performa basis data operasional agar tidak terganggu oleh permintaan laporan yang bersifat berat.

Dari sisi backend, perubahan mekanisme pemrosesan dari sinkron menjadi asinkron memberikan kontribusi besar terhadap peningkatan responsivitas dan stabilitas sistem. Sistem tidak lagi menunggu seluruh proses pengambilan dan pengolahan data selesai sebelum memberikan respons kepada pengguna. Pendekatan ini mendukung temuan Joshi dan Zulfa et al., yang menyatakan bahwa pemrosesan asinkron dan pemanfaatan *in-memory caching* dapat meningkatkan keandalan sistem dan mengurangi risiko kegagalan respons pada aplikasi berbasis web [7], [8].

Integrasi Redis sebagai mekanisme pencatatan status dan progres pembuatan laporan memungkinkan sistem memberikan umpan balik secara real-time kepada pengguna. Selain meningkatkan transparansi proses, mekanisme ini juga mencegah eksekusi laporan yang sama secara bersamaan, sehingga menghindari pemborosan sumber daya sistem. Hasil penelitian ini memperkuat pandangan bahwa Redis tidak hanya efektif sebagai media caching data, tetapi juga sebagai komponen pendukung pengelolaan proses backend pada sistem berskala besar.

Pemanfaatan MinIO sebagai media penyimpanan hasil laporan dalam format JSON turut berkontribusi dalam meningkatkan efisiensi sistem pelaporan. Dengan menyimpan hasil laporan yang telah selesai diproses, sistem dapat melayani permintaan laporan yang sama tanpa perlu menjalankan ulang proses pengolahan data. Pendekatan ini sejalan dengan penelitian Malhotra dan Saqib serta Zhang et al., yang menunjukkan bahwa *object storage* efektif digunakan untuk menyimpan data tidak terstruktur dan mendukung sistem dengan kebutuhan akses data berulang [9], [10].

Secara keseluruhan, pembahasan hasil menunjukkan bahwa optimalisasi sistem pelaporan penunjang medis melalui pendekatan terintegrasi pada level basis data dan backend mampu menjawab permasalahan performa dan stabilitas yang sebelumnya terjadi. Pendekatan ini tidak hanya relevan secara teknis, tetapi juga aplikatif untuk diterapkan pada sistem informasi rumah sakit yang memiliki karakteristik data besar dan kebutuhan layanan yang bersifat kritis.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan strategi optimalisasi kinerja pada sistem pelaporan penunjang medis di lingkungan Sistem Informasi Manajemen Rumah Sakit melalui pendekatan terintegrasi pada level basis data dan backend. Optimalisasi dilakukan dengan menerapkan *indexing* dan *materialized view* pada basis data, pemrosesan backend secara asinkron, serta integrasi Redis dan MinIO sebagai mekanisme *caching*, pelacakan progres, dan penyimpanan hasil laporan. Pendekatan ini dirancang untuk mengatasi permasalahan keterlambatan respons dan ketidakstabilan sistem pada proses pelaporan dengan volume data besar.

Hasil pengujian menunjukkan bahwa optimalisasi yang diterapkan memberikan peningkatan kinerja yang sangat signifikan. Pada modul Laboratorium dan Radiologi, waktu respons sistem mengalami peningkatan rata-rata lebih dari 99%, sedangkan pada modul Patologi Anatomi peningkatan kinerja mencapai lebih dari 96%. Selain itu, penerapan pemrosesan asinkron dan mekanisme pencatatan status berbasis Redis berhasil menghilangkan kegagalan respons seperti *Gateway Timeout* (HTTP 504) yang sebelumnya sering terjadi. Temuan ini membuktikan bahwa kombinasi optimasi basis data dan arsitektur

backend yang tepat mampu meningkatkan responsivitas dan stabilitas sistem secara bersamaan.

Dari sisi sistem, optimalisasi ini berhasil mengurangi beban eksekusi query berat pada basis data operasional dan meningkatkan efisiensi pemrosesan laporan. Pemanfaatan MinIO sebagai media penyimpanan hasil laporan dalam format JSON memungkinkan penggunaan kembali laporan tanpa perlu pemrosesan ulang, sehingga mendukung skalabilitas sistem pelaporan. Dari sisi pengguna, sistem yang dihasilkan memberikan waktu respons yang jauh lebih cepat, transparansi proses melalui informasi progres pembuatan laporan, serta pengalaman penggunaan yang lebih andal dalam mendukung aktivitas operasional rumah sakit.

Meskipun hasil penelitian menunjukkan peningkatan kinerja yang signifikan, penelitian ini memiliki beberapa keterbatasan. Optimalisasi diuji pada modul penunjang medis dengan karakteristik data tertentu dan menggunakan rentang waktu pengujian yang terbatas. Selain itu, evaluasi kinerja difokuskan pada waktu respons dan stabilitas sistem, tanpa membahas aspek penggunaan sumber daya seperti konsumsi memori dan CPU secara mendalam. Penelitian

DAFTAR PUSTAKA

- [1] R. N. Azizah, "PENERAPAN SISTEM INFORMASI MANAJEMEN RUMAH SAKIT (SIMRS) DALAM MENINGKATKAN EFISIENSI PELAYANAN KESEHATAN DI INDONESIA : STUDI LITERATUR," vol. 6, pp. 6667–6674, 2025.
- [2] C. Jurnal et al., "Hubungan Implementasi Sistem Informasi Manajemen Rumah Sakit (SIMRS) dengan Kinerja Perawat Universitas Islam Sultan Agung , Indonesia keperawatan , kepala departemen , perawat , dan seluruh pengambil keputusan dalam organisasi . menerapkan komunikasi terapeutik , maka berdampak pada rendahnya kepuasan pasien .," vol. 3, 2025.
- [3] Wikipedia, "Responsiveness," Wikipedia. Accessed: Dec. 20, 2025. [Online]. Available: <https://en.wikipedia.org/wiki/Responsiveness>
- [4] M. Abbasi et al., "Revisiting Database Indexing for Parallel and Accelerated Computing: A Comprehensive Study and Novel Approaches".
- [5] S. Sakshi and A. Sharma, *Relational Database Performance Optimization Techniques*. Atlantis Press International BV, 2025. doi: 10.2991/978-94-6463-716-8.
- [6] V. S. Ramdoss, "Optimizing database queries : Cost and performance analysis," vol. 02, no. May, pp. 293–297, 2021.
- [7] M. I. Zulfa, A. Fadli, A. W. Wardhana, A. Fadli, and A. W. Wardhana, "Strategi caching aplikasi berbasis in-memory menggunakan Redis server untuk mempercepat akses data relasional Application caching strategy based on in-memory using Redis server to accelerate," vol. 8,

- no. June 2019, pp. 157–163, 2020, doi: 10.14710/jtsiskom.8.2.2020.157-163.
- [8] P. K. Joshi, “Redis Cache Optimization for Payment Gateways in the Cloud,” no. June 2023, 2025, doi: 10.63282/30509246/IJETCSIT-V4I2P104.
- [9] S. Malhotra and M. Saqib, “Evaluating Fault Tolerance and Scalability in Distributed File Systems : A Case Study of GFS ,”.
- [10] W. Zhang, C. Li, and J. Duan, “Research on the Method of Massive Data Storage Management,” no. 17, 2024, doi: 10.4108/eai.27-10-2023.2342009.
- [11] A. M. Bonteanu and C. Tudose, “Performance Analysis and Improvement for CRUD Operations in Relational Databases from Java Programs Using JPA , Hibernate , Spring Data JPA”.
- [12] A. A. Albert and N. G. Blas, “Intelligent Indexing — Boosting Performance in Database Applications by Recognizing Index Patterns,” pp. 1–14, 2020, doi: 10.3390/electronics9091348.
- [13] M. R. Putra, A. Hasan, V. Stefanus, A. B. Maghribi, M. Irawan, and R. Menzona, “PENGARUH INDEXED VIEWS TERHADAP PERFORMA QUERY PADA BASIS DATA SQL SERVER DENGAN DATA SKALA BESAR,” vol. 9, no. 2, pp. 2586–2591, 2025, doi: <https://doi.org/10.36040/jati.v9i2.13159>.
- [14] S. C. Hubli and R. C. Jaiswal, “Efficient Backend Development with Spring Boot :,” no. November, 2023.
- [15] M. V. Privalov and M. V. Stupina, “Improving web-oriented information systems efficiency using Redis caching mechanisms,” vol. 33, no. 3, pp. 1667–1675, 2024, doi: 10.11591/ijeecs.v33.i3.pp1667-1675.
- [16] P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, “API Testing Using Postman Tool” no. December, 2022.