

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi mendorong perusahaan dan organisasi untuk mengelola data secara terintegrasi, cepat, dan akurat. Salah satu sistem yang banyak mendukung kebutuhan tersebut adalah ERP atau *Enterprise Resource Planning*. Sistem ERP berperan penting dalam mengintegrasikan banyak proses bisnis seperti keuangan, sumber daya manusia, logistik dan operasional dalam satu platform terpusat. Seluruh aktivitas tersebut direpresentasikan dalam bentuk data transaksi yang diproses secara kontinu.

Dalam implementasinya, sistem ERP sangat bergantung pada kinerja sistem manajemen basis data karena seluruh proses bisnis menghasilkan data transaksi yang harus dikelola secara konsisten. Meskipun terdapat berbagai model basis data, sistem ERP umumnya lebih sesuai menggunakan relational database management system (RDBMS) karena mampu menjaga struktur data, relasi antar entitas, integritas data, serta mendukung transaksi ACID yang penting bagi proses bisnis operasional.

Salah satu RDBMS yang banyak digunakan adalah PostgreSQL. PostgreSQL merupakan sistem basis data open-source yang mendukung transaksi ACID, concurrency control, serta kemampuan pemrosesan query yang dapat digunakan untuk kebutuhan transaksi maupun analisis data. Karakteristik tersebut menjadikan PostgreSQL relevan untuk digunakan dalam sistem ERP, khususnya pada organisasi yang membutuhkan basis data relasional yang fleksibel dan tidak bergantung pada solusi proprietary.

Dalam operasionalnya, sistem ERP berbasis PostgreSQL umumnya melayani dua jenis beban kerja utama yaitu *online transaction processing* (OLTP) dan *online analytical processing* (OLAP). Beban kerja OLTP berfokus pada pemrosesan transaksi harian seperti *insert*, *update*, dan *select* sederhana yang membutuhkan waktu respon rendah dan *throughput* stabil. Sebaliknya,

beban kerja OLAP melibatkan query kompleks seperti agregasi, join antar tabel, dan analisis historis yang memerlukan pemrosesan data dalam jumlah besar serta waktu eksekusi query yang lebih panjang.

Relevansi PostgreSQL dalam konteks ERP juga dapat dilihat pada studi benchmarking ERP open-source. Dalam penelitian tersebut konfigurasi pengujianya menggunakan PostgreSQL 9.4 sebagai salah satu kebutuhan server. Studi tersebut juga menekankan bahwa ERP dipilih sebagai objek evaluasi karena memiliki banyak transaksi, modul yang beragam, serta kebutuhan pengujian dengan jumlah data dan beban pengguna konkuren yang besar [[1]].

Perbedaan karakteristik OLTP dan OLAP menyebabkan keduanya memiliki pola akses data dan pola pemrosesan query yang berbeda. OLTP cenderung sensitif terhadap *latency* dan membutuhkan eksekusi transaksi yang cepat dan konsisten, sedangkan OLAP lebih menekankan pada pemrosesan query analitik dan *throughput* analitik. Dalam praktik implementasi ERP, terutama pada organisasi skala kecil dan menengah, kedua jenis beban kerja tersebut sering dijalankan secara bersamaan dalam satu basis data yang sama tanpa pemisahan arsitektur seperti *data warehouse* atau sistem *hybrid transactional and analytical processing* (HTAP).

Di sisi lain, penelitian sebelumnya menunjukkan bahwa eksekusi beban kerja OLTP dan OLAP secara bersamaan dapat menimbulkan interferensi beban kerja. Salah satu studi terkait HTAP melaporkan bahwa *throughput* OLTP dapat menurun hingga 42% ketika dijalankan bersamaan dengan OLAP akibat kontensi *cache* dan *bandwidth* memori [[2]]. Temuan tersebut menunjukkan bahwa *mixed workload* berpotensi memengaruhi stabilitas kinerja transaksi, sehingga perlu dikaji lebih lanjut pada konteks ERP berbasis PostgreSQL.

Kondisi *coexistence* tersebut berpotensi menimbulkan interferensi beban kerja atau *workload interference*, yaitu kondisi ketika eksekusi query analitik yang kompleks berjalan bersamaan dengan transaksi operasional dalam satu basis data. Interferensi tersebut dapat terlihat melalui perubahan kinerja OLTP,

seperti peningkatan response time dan penurunan throughput transaksi. Beberapa penelitian menunjukkan bahwa eksekusi OLAP secara bersamaan dengan OLTP dapat menyebabkan kontensi *cache processor*, saturasi *bandwidth* memori, serta peningkatan *cache miss* yang berdampak pada penurunan *throughput* transaksi dan peningkatan waktu respon OLTP [2], [3], [4]. Interferensi ini menjadi perhatian karena stabilitas transaksi merupakan kebutuhan utama dalam sistem ERP.

Berbagai penelitian telah mengusulkan pendekatan untuk mengatasi permasalahan tersebut, terutama melalui arsitektur HTAP yang memisahkan atau mengisolasi sumber daya antara beban kerja transaksi dan analitik. Studi eksperimental menunjukkan bahwa tanpa mekanisme isolasi yang memadai, *throughput* OLTP dapat mengalami penurunan signifikan ketika dijalankan bersamaan dengan OLAP [2], [3]. Di sisi lain, pengembangan sistem HTAP dan *benchmark hybrid* bertujuan untuk meminimalkan dampak interferensi tersebut melalui desain arsitektur khusus.

Namun demikian, sintesis terhadap penelitian terdahulu menunjukkan bahwa kajian mengenai interaksi OLTP dan OLAP masih memiliki beberapa keterbatasan dalam konteks penelitian ini. Penelitian terkait benchmark HTAP, seperti OLxPBench, telah menekankan pentingnya evaluasi beban kerja campuran melalui skema transaksi, query analitik, dan hybrid workload, tetapi fokusnya masih berada pada perancangan benchmark dan evaluasi lintas sistem, bukan pada sistem ERP berbasis PostgreSQL secara spesifik [4]. Penelitian lain terkait sistem HTAP cloud-native, seperti PolarDB-IMCI, menunjukkan bahwa mekanisme isolasi sumber daya dan arsitektur khusus dapat mengurangi dampak beban kerja analitik terhadap transaksi, tetapi konteks tersebut berbeda dengan sistem ERP konvensional yang berjalan pada satu basis data PostgreSQL tanpa mekanisme HTAP khusus [3].

Penelitian lain juga belum sepenuhnya menjawab fokus penelitian ini. Studi mengenai akselerasi GPU pada DBMS lebih menitikberatkan pada peningkatan throughput OLTP melalui pemrosesan berbasis CPU-GPU, sehingga belum membahas interaksi beban kerja OLTP dan OLAP dalam satu basis data ERP

[5]. Sementara itu, penelitian mengenai OLAP multidimensi lebih berfokus pada pemodelan data mart dan analisis data historis, bukan pada pengaruh eksekusi query OLAP terhadap kinerja OLTP [6]. Penelitian interferensi HTAP memang menunjukkan bahwa eksekusi OLTP dan OLAP secara bersamaan dapat menurunkan throughput transaksi, tetapi kajiannya belum diarahkan pada sistem ERP berbasis PostgreSQL dengan skenario OLAP ringan dan OLAP berat [2].

Berdasarkan sintesis tersebut, research gap penelitian ini terletak pada belum adanya kajian eksperimental yang secara spesifik mengukur pengaruh beban kerja OLAP ringan dan OLAP berat terhadap kinerja OLTP pada sistem ERP berbasis PostgreSQL dalam satu basis data yang sama. Penelitian ini berbeda dari penelitian sebelumnya karena tidak berfokus pada rancangan arsitektur HTAP, benchmark lintas sistem, akselerasi perangkat keras, atau pemodelan OLAP secara terpisah, melainkan pada pengukuran dampak mixed workload terhadap response time, throughput, dan statistik level database melalui `pg_stat_statements`. Dengan demikian, penelitian ini diarahkan untuk memberikan bukti empiris pada konteks ERP PostgreSQL tanpa pemisahan arsitektural seperti data warehouse atau HTAP.

Berdasarkan kondisi tersebut, diperlukan kajian empiris untuk mengukur dan menganalisis dampak eksekusi beban kerja OLAP terhadap kinerja OLTP dalam lingkungan ERP berbasis PostgreSQL yang terkontrol. Penelitian ini bertujuan untuk mengisi celah tersebut dengan mengevaluasi karakteristik beban kerja OLTP dan OLAP serta mengukur tingkat interferensi yang terjadi ketika keduanya dijalankan secara simultan dalam satu basis data.

Secara spesifik, penelitian ini memberikan kontribusi berupa evaluasi eksperimental terhadap dampak beban kerja OLAP terhadap kinerja OLTP pada sistem ERP berbasis PostgreSQL yang dijalankan dalam satu basis data tanpa pemisahan arsitektur. Evaluasi dilakukan pada tiga perangkat, yaitu laptop Dell, HP ProDesk, dan MacBook, melalui tiga skenario pengujian yang meliputi baseline OLTP-only, mixed workload OLAP ringan, dan mixed workload OLAP berat. Metrik utama yang digunakan adalah response time dan throughput OLTP, sedangkan statistik level database melalui

pg_stat_statements digunakan sebagai informasi pendukung. Hasil pengujian dianalisis menggunakan rata-rata, standar deviasi, persentase perubahan kinerja, One-Way ANOVA, dan Two-Way ANOVA, sehingga dampak keberadaan beban kerja OLAP terhadap kinerja transaksi dapat dianalisis secara kuantitatif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini merumuskan beberapa permasalahan yang akan dikaji melalui pendekatan eksperimental. Rumusan masalah tersebut adalah sebagai berikut:

1. Bagaimana kinerja beban kerja OLTP pada sistem ERP berbasis PostgreSQL dalam kondisi baseline OLTP-only berdasarkan metrik response time, throughput, dan statistik level database?
2. Bagaimana karakteristik beban kerja OLAP ringan dan OLAP berat yang digunakan dalam skenario mixed workload berdasarkan kompleksitas query, jumlah join, agregasi, dan waktu eksekusi query?
3. Bagaimana pengaruh beban kerja OLAP ringan dan OLAP berat terhadap kinerja OLTP pada sistem ERP berbasis PostgreSQL berdasarkan perbandingan response time, throughput, persentase perubahan kinerja, standar deviasi, serta hasil uji One-Way ANOVA dan Two-Way ANOVA?

1.3 Batasan Masalah

Penelitian ini dibatasi pada analisis pengaruh beban kerja OLAP terhadap kinerja OLTP pada sistem ERP yang menggunakan PostgreSQL sebagai basis data utama. Pengujian dilakukan pada satu basis data yang sama pada masing-masing perangkat tanpa menerapkan pemisahan arsitektur data seperti data warehouse atau sistem hybrid transactional and analytical processing (HTAP). Beban kerja OLTP yang dianalisis mencakup operasi baca dan tulis data transaksional yang merepresentasikan aktivitas akses dan pemrosesan data pada sistem ERP.

Operasi OLTP dalam penelitian ini meliputi SELECT, INSERT, UPDATE, dan DELETE. Workload OLTP dijalankan sebanyak 50.000 iterasi per run, dengan komposisi 6 operasi SELECT, 1 operasi INSERT, 1 operasi UPDATE, dan 1 operasi DELETE pada setiap iterasi, sehingga total operasi OLTP yang dijalankan

adalah 450.000 operasi per run. Operasi SELECT dijalankan pada tabel utama ERP, sedangkan operasi INSERT, UPDATE, dan DELETE dilakukan pada tabel khusus pengujian agar tidak mengubah data utama ERP yang digunakan dalam query OLAP.

Beban kerja OLAP dibatasi pada eksekusi query analitik berupa agregasi data, join antar tabel, dan perhitungan laporan yang mencerminkan kebutuhan analisis data pada sistem ERP. Beban kerja OLAP dikelompokkan menjadi OLAP ringan dan OLAP berat berdasarkan kompleksitas query, jumlah join, agregasi, dan waktu eksekusi query.

Metrik utama yang digunakan untuk menilai kinerja OLTP adalah response time dan throughput. Penelitian ini juga menggunakan statistik level database melalui `pg_stat_statements`, yaitu `total_exec_time`, `disk_read`, dan `cache_hit`, sebagai informasi pendukung dalam analisis hasil pengujian. Penelitian ini tidak melakukan monitoring penggunaan sumber daya pada tingkat sistem operasi seperti CPU, RAM, maupun disk I/O OS.

Penelitian ini tidak membahas teknik optimasi database, pengaturan konfigurasi lanjutan pada PostgreSQL, maupun evaluasi solusi atau pendekatan arsitektural tertentu. Data yang digunakan dalam penelitian ini merupakan data uji non-produksi yang disesuaikan untuk kebutuhan pengujian.

Pengujian dilakukan pada tiga perangkat berbeda, yaitu laptop Dell, HP ProDesk, dan MacBook, dengan skema basis data ERP, dataset, query workload, jumlah operasi, dan prosedur pengujian yang dibuat sama. Setiap skenario pengujian dilakukan sebanyak lima kali pengulangan pada masing-masing perangkat. Penggunaan tiga perangkat bertujuan untuk memperluas variasi lingkungan pengujian, namun hasil penelitian tetap berfokus pada pengamatan pola pengaruh beban kerja OLAP terhadap kinerja OLTP dalam kondisi eksperimen yang telah ditentukan, bukan untuk menentukan perangkat dengan kinerja terbaik secara absolut. Oleh karena itu, temuan yang diperoleh tidak dimaksudkan untuk digeneralisasi secara langsung ke seluruh jenis sistem, perangkat, lingkungan produksi, atau konfigurasi PostgreSQL lainnya.

1.4 Tujuan dan Manfaat Penelitian

Tujuan dan manfaat penelitian dalam penelitian ini merupakan bagian yang menjelaskan arah pelaksanaan penelitian serta kontribusi yang diharapkan dari hasil yang diperoleh. Adapun tujuan dan manfaat penelitian ini sebagai berikut.

1.4.1 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan, penelitian ini memiliki beberapa tujuan yang ingin dicapai melalui pendekatan eksperimental. Tujuan penelitian tersebut adalah sebagai berikut:

1. Menganalisis kinerja beban kerja OLTP pada sistem ERP berbasis PostgreSQL dalam kondisi baseline OLTP-only berdasarkan metrik response time, throughput, dan statistik level database.
2. Menganalisis karakteristik beban kerja OLAP ringan dan OLAP berat pada sistem ERP berbasis PostgreSQL berdasarkan kompleksitas query, jumlah join, agregasi, dan waktu eksekusi query.
3. Mengevaluasi pengaruh eksekusi beban kerja OLAP ringan dan OLAP berat terhadap kinerja OLTP ketika dijalankan secara bersamaan dalam satu basis data PostgreSQL berdasarkan response time, throughput, persentase perubahan kinerja, standar deviasi, serta hasil uji One-Way ANOVA dan Two-Way ANOVA.

1.4.2 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan bukti empiris mengenai pengaruh beban kerja OLAP terhadap kinerja OLTP pada sistem ERP berbasis PostgreSQL dalam satu lingkungan basis data yang sama.
2. Menjadi referensi akademik terkait karakteristik interaksi antara beban kerja OLTP dan OLAP pada sistem basis data PostgreSQL tanpa pemisahan arsitektur data.

3. Menyediakan hasil pengujian eksperimental mengenai perubahan response time, throughput, dan statistik level database ketika beban kerja OLTP dan OLAP dijalankan secara bersamaan.
4. Menjadi dasar pemahaman awal bagi penelitian selanjutnya yang membahas optimasi, tuning, atau pendekatan arsitektural untuk mengurangi dampak beban kerja OLAP terhadap kinerja OLTP pada sistem ERP berbasis PostgreSQL.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini disusun untuk memberikan gambaran yang terstruktur mengenai isi setiap bab dalam laporan skripsi. Adapun sistematika penulisan ini adalah sebagai berikut

BAB I PENDAHULUAN

- Bab ini berisi latar belakang penelitian yang menjelaskan konteks dan urgensi penelitian ini. Selain itu, bab ini juga memuat rumusan masalah, batasan masalah, tujuan, dan manfaat penelitian, serta sistematika penulisan sebagai kerangka penyusunan laporan.

BAB II LANDASAN TEORI

- Bab ini membahas teori-teori yang mendukung penelitian, meliputi konsep *Enterprise resource planning*, arsitektur PostgreSQL, karakteristik beban kerja OLTP dan OLAP, mekanisme manajemen sumber daya dan locking, serta metrik kinerja sistem seperti *throughput*, *response time (latency)*, dan statistik level database. Bab ini juga memuat tinjauan penelitian terdahulu yang relevan.

BAB III METODOLOGI PENELITIAN

- Bab ini menjelaskan metode penelitian yang digunakan, dan juga pembahasan gambaran umum objek penelitian, arsitektur sistem ERP yang diuji, lingkungan pengujian, perancangan eksperimen dan skenario pengujian, teknik pengumpulan data, variabel penelitian, perancangan

beban kerja, dan juga Teknik analisis data yang digunakan untuk mengevaluasi hasil pengujian.

BAB IV ANALISIS DAN HASIL PENELITIAN

- Bab ini menyajikan hasil implementasi dan pengujian eksperimen berdasarkan skenario yang telah dirancang, untuk menjawab rumusan masalah yang telah disusun. Hasil pengujian dianalisis dalam bentuk perhitungan numerik dan visualisasi grafik untuk mengidentifikasi tingkat interferensi serta dampaknya terhadap *response time*, *throughput*, dan statistik level database.

BAB V SIMPULAN DAN SARAN

- Bab ini berisi simpulan yang diperoleh berdasarkan hasil analisis dan pembahasan penelitian. Simpulan menjawab rumusan masalah yang telah ditetapkan pada Bab I. Selain itu, bab ini juga memuat saran untuk pengembangan penelitian selanjutnya.

