



Optimasi Kueri pada Very Large Database untuk Analisis Transaksi E-Commerce Real-Time Menggunakan Hybrid Indexing dan Partitioning

***Candra Purnama**

Program Studi Sistem Informasi, Fakultas Teknik, Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI, Indonesia

*Email: cpillahi@gmail.com

Abstract

The growth rate of e-commerce transactions generates enormous data volumes (Very Large Database/VLDB), creating significant performance challenges for real-time analytical queries. Traditional optimization techniques are often inadequate when applied separately. This study proposes and tests a hybrid optimization strategy that integrates Range Partitioning and Partial Indexing in a PostgreSQL database system. An experimental research method was conducted by building a database simulation containing 100 million rows of synthetic transaction data, then comparing performance between a baseline configuration and an optimized configuration. Test results show significant improvement. The complex aggregation query (Q1) experienced an 86.6% acceleration in execution time (from 14,200 ms to 1,900 ms), while the specific search query (Q2) improved by 89.1% (from 8,500 ms to 930 ms). Query plan analysis proves the effectiveness of the partition pruning mechanism and the use of more efficient partial indexes. It is concluded that this hybrid strategy is effective in optimizing VLDB performance for e-commerce analytical workloads and is recommended for adoption in production environments.

Keywords: Very Large Database, Query Optimization, Partitioning, Indexing, E-commerce, PostgreSQL.

Abstrak

Laju pertumbuhan transaksi e-commerce menghasilkan volume data yang sangat besar (Very Large Database/VLDB), menciptakan tantangan performa signifikan terhadap kueri analitik real-time. Teknik optimasi tradisional sering kali tidak memadai secara terpisah. Penelitian ini bertujuan mengusulkan dan menguji strategi optimasi hybrid yang mengintegrasikan Range Partitioning dan Partial Indexing pada sistem basis data PostgreSQL. Metode penelitian eksperimental dilakukan dengan membangun simulasi database yang berisi 100 juta baris data transaksi sintesis, kemudian membandingkan performa antara konfigurasi baseline dengan konfigurasi yang telah dioptimasi. Hasil pengujian menunjukkan peningkatan yang signifikan. Kueri agregasi kompleks (Q1) mengalami percepatan waktu eksekusi sebesar 86.6% (dari 14,200 ms menjadi 1,900 ms), sementara kueri pencarian spesifik (Q2) meningkat 89.1% (dari 8,500 ms menjadi 930 ms). Hasil analisis query plan membuktikan efektivitas mekanisme partition pruning dan penggunaan indeks parsial yang lebih efisien. Disimpulkan bahwa strategi hybrid ini efektif mengoptimalkan performa VLDB untuk workload analitik e-commerce dan direkomendasikan untuk diadopsi dalam lingkungan produksi.

Kata Kunci: Very Large Database, Optimasi Kueri, Partitioning, Indexing, E-Commerce, PostgreSQL.



PENDAHULUAN

Ekonomi digital Indonesia telah mengalami transformasi luar biasa, dengan sektor e-commerce sebagai salah satu penggerak utamanya. Platform seperti Shopee, Tokopedia, dan Bukalapak tidak hanya memfasilitasi jutaan transaksi harian tetapi juga menghasilkan *data footprint* yang berskala sangat besar (*Very Large Database* atau VLDB). Data ini menjadi aset kritis untuk analisis bisnis real-time, seperti pelaporan penjualan, analisis tren konsumen, dan personalisasi rekomendasi. Namun, volume data yang telah mencapai orde ratusan gigabyte hingga terabyte ini menciptakan tantangan performa yang kompleks. Kueri analitik yang melibatkan operasi *join*, agregasi, dan filter temporal pada tabel dengan miliaran baris sering kali mengalami *latency* tinggi, menghambat pengambilan keputusan yang gesit.

Pada lingkungan VLDB, pendekatan optimasi konvensional menunjukkan keterbatasan. Pembuatan indeks (*indexing*) konvensional pada semua kolom meningkatkan biaya penyimpanan dan memperlambat operasi penulisan (*write*). Di sisi lain, membagi tabel (*partitioning*) tanpa strategi indeks yang tepat belum tentu memberikan keuntungan optimal. Penelitian ini berangkat dari identifikasi kesenjangan bahwa kedua teknik ini jarang didesain secara sinergis untuk *workload* spesifik e-commerce, yang ditandai dengan pola akses yang tidak merata data terbaru (*hot data*) diakses jauh lebih intensif daripada data historis. Oleh karena itu, penelitian ini dirancang untuk menjawab pertanyaan utama: Bagaimana efektivitas strategi optimasi *hybrid* yang mengintegrasikan **Range Partitioning** berbasis waktu dengan **Partial Indexing** selektif dalam meningkatkan performa kueri analitik pada VLDB simulasi transaksi e-commerce?

Tujuan penelitian adalah untuk merancang, mengimplementasikan, dan mengukur dampak strategi tersebut terhadap waktu eksekusi kueri. Kontribusi penelitian ini adalah memberikan solusi praktis berbasis bukti (*evidence-based*) yang dapat diadopsi oleh pengembang dan *Database Administrator* (DBA) untuk mengoptimalkan infrastruktur data dalam menghadapi ledakan data e-commerce di Indonesia.

KAJIAN TEORITIS

Konsep Very Large Database (VLDB) dan Big Data dalam E-Commerce

Very Large Database (VLDB) merujuk pada sistem basis data yang menangani volume, kecepatan, dan variasi data di luar kemampuan sistem konvensional. Dalam konteks e-commerce, big data dicirikan oleh 3V: Volume (skala data transaksi dan log), Velocity (kecepatan data masuk real-time), dan Variety (data terstruktur seperti transaksi dan data tidak terstruktur seperti ulasan). Data ini bernilai tinggi untuk analisis perilaku konsumen, manajemen inventori, dan prediksi tren pasar.

Partitioning, teknik membagi tabel logis menjadi segmen fisik lebih kecil (partisi) berdasarkan kunci tertentu, seperti rentang waktu (Range Partitioning). Keunggulan utamanya adalah partition pruning, di mana optimizer basis data hanya memindai partisi yang relevan dengan kondisi kueri, secara drastis mengurangi **I/O**.

Indexing, struktur data (seperti B-Tree) yang mempercepat pencarian. Pada VLDB, indeks lengkap menimbulkan overhead tinggi. Partial Indexing adalah solusi dengan membuat indeks hanya untuk subset data (misal, data 6 bulan terakhir), mengurangi ukuran indeks dan biaya pemeliharannya.

Workload sistem e-commerce bersifat hybrid, mencakup operasi transaksional (OLTP) dan analitik (OLAP). Untuk query analitik, pola aksesnya sangat dipengaruhi oleh waktu, di mana kueri laporan harian/bulanan dan analisis pelanggan selalu menyertakan filter temporal. Pola ini membuka peluang optimal untuk partitioning berbasis waktu dan indexing selektif pada data aktif.

METODE PENELITIAN

Penelitian ini dirancang dengan pendekatan eksperimental kuantitatif untuk menguji hipotesis bahwa integrasi strategis antara teknik partitioning dan indexing dapat menghasilkan peningkatan performa yang lebih besar dibandingkan penerapannya secara terpisah pada lingkungan VLDB. Eksperimen dilakukan dalam lingkungan yang terkendali dengan membandingkan secara langsung dua skenario konfigurasi basis data terhadap serangkaian workload kueri yang terstandarisasi.

Proses penelitian diawali dengan identifikasi masalah yang berfokus pada analisis bottleneck performance pada sistem Very Large Database (VLDB) dalam lingkungan e-commerce, dengan tujuan menemukan faktor-faktor utama yang menyebabkan penurunan kinerja sistem. Tahap berikutnya adalah studi literatur dan perancangan, yang mencakup kajian terhadap teknik partitioning dan indexing serta perancangan skema basis data dan strategi hybrid sebagai solusi konseptual untuk meningkatkan performa.

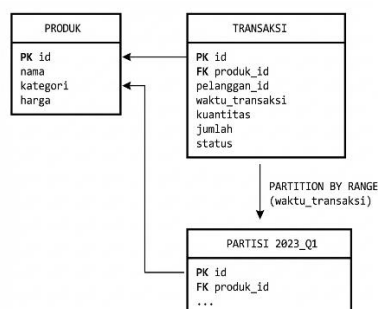
Setelah perancangan selesai, penelitian dilanjutkan pada tahap implementasi dan persiapan data dengan menyiapkan lingkungan database PostgreSQL sebagai platform pengujian. Pada tahap ini dilakukan pembuatan data sintetis berskala besar sebanyak 150 juta baris untuk merepresentasikan kondisi VLDB, serta implementasi dua skema database, yaitu skema A sebagai baseline dan skema B sebagai skema hybrid. Tahap pengujian performansi kemudian dilakukan dengan mengeksekusi query uji sebanyak 10 dan 20 kali, disertai pengukuran waktu eksekusi serta analisis query plan, dengan pengulangan eksperimen sebanyak 15 iterasi untuk memastikan konsistensi hasil.

Tahap selanjutnya adalah analisis data dan evaluasi, di mana hasil pengukuran dianalisis secara statistik untuk menilai perbedaan kinerja antar skema serta mengevaluasi efektivitas strategi hybrid yang diterapkan. Berdasarkan hasil analisis tersebut, penelitian diakhiri dengan penarikan kesimpulan yang merangkum temuan utama serta penyusunan rekomendasi yang bersifat implementatif dan akademis, baik untuk penerapan sistem database berskala besar maupun untuk pengembangan penelitian selanjutnya. (1) Identifikasi Masalah & Studi Literatur, (2) Perancangan Skema & Strategi Hybrid, (3) Implementasi Skenario (A: Baseline, B: Hybrid), (4) Pembangkitan & Loading Data, (5) Eksekusi Kueri Uji & Pengukuran, (6) Analisis Data & Evaluasi.

HASIL

Perancangan Lingkungan dan Data Uji

Lingkungan eksperimen dibangun pada server dengan spesifikasi : Intel Xeon 8-core, RAM 32 GB, SSD NVMe, OS Ubuntu 20.04 LTS dan PostgreSQL 14. Penelitian ini menggunakan model database relasional dengan dua table utama. Diagram Entity-Relationship (ERD) system ditunjukkan pada Gambar 1.



Gambar 1: Diagram Entity-Relationship Database

Tabel 1. Struktur Tabel produk (Tabel Dimensi)

Kolom	Tipe Data	Deskripsi	Constraints
id	BIGSERIAL	Identifikasi unik produk	PRIMARY KEY
nama	VARCHAR(255)	Nama produk	NOT NULL
kategori	VARCHAR(100)	Kategori produk	NOT NULL
harga	DECIMAL(12,2)	Harga satuan produk	NOT NULL

Tabel 2. Struktur Tabel transaksi (Tabel Fakta)

Kolom	Tipe Data	Deskripsi	Constraints
id	BIGSERIAL	Identifikasi unik transaksi	PRIMARY KEY
produk_id	BIGINT	Referensi ke tabel produk	FOREIGN KEY
pelanggan_id	BIGINT	Identifikasi pelanggan	NOT NULL
waktu_transaksi	TIMESTAMP	Waktu terjadinya transaksi	NOT NULL
kuantitas	INTEGER	Jumlah barang	NOT NULL
jumlah	DECIMAL(12,2)	Total nilai transaksi	NOT NULL
status	VARCHAR(20)	Status transaksi	CHECK values

Skenario Optimasi

Pada skenario A (baseline – tanpa optimasi) ini, tabel **transaksi** diimplementasikan sebagai single table tanpa partitioning:

```
CREATE TABLE transaksi (
  id BIGSERIAL PRIMARY KEY,
  produk_id BIGINT REFERENCES produk(id),
  pelanggan_id BIGINT NOT NULL,
  waktu_transaksi TIMESTAMP NOT NULL,
  kuantitas INTEGER NOT NULL,
```

```
    jumlah DECIMAL(12,2) NOT NULL,  
    status VARCHAR(20) CHECK (status IN ('sukses', 'gagal', 'dibatalkan'))  
);
```

Hanya indeks default pada kolom id yang dibuat. Tidak ada indeks tambahan pada kolom lainnya. Pada scenario B (strategi hybrid) ini, diterapkan kombinasi range partitioning dan partial indexing:

Implementasi Range Partitioning

```
CREATE TABLE transaksi (  
    id BIGSERIAL,  
    produk_id BIGINT NOT NULL,  
    pelanggan_id BIGINT NOT NULL,  
    waktu_transaksi TIMESTAMP NOT NULL,  
    kuantitas INTEGER NOT NULL,  
    jumlah DECIMAL (12,2) NOT NULL,  
    status VARCHAR (20),  
    PRIMARY KEY (id, waktu_transaksi)  
) PARTITION BY RANGE (waktu_transaksi);
```

```
CREATE TABLE transaksi_2023_q1 PARTITION OF transaksi  
    FOR VALUES FROM ('2023-01-01') TO ('2023-04-01');
```

```
CREATE TABLE transaksi_2023_q2 PARTITION OF transaksi  
    FOR VALUES FROM ('2023-04-01') TO ('2023-07-01');
```

```
CREATE TABLE transaksi_2023_q3 PARTITION OF transaksi  
    FOR VALUES FROM ('2023-07-01') TO ('2023-10-01');
```

```
CREATE TABLE transaksi_2023_q4 PARTITION OF transaksi  
    FOR VALUES FROM ('2023-10-01') TO ('2024-01-01');
```

Implementasi Partial Indexing

-- Partial index hanya untuk data tahun 2023

```
CREATE INDEX idx_transaksi_produk_id_2023 ON transaksi(produk_id)  
    WHERE waktu_transaksi >= '2023-01-01';
```

```
CREATE INDEX idx_transaksi_pelanggan_id_2023 ON transaksi(pelanggan_id)  
    WHERE waktu_transaksi >= '2023-01-01';
```

```
CREATE INDEX idx_transaksi_waktu_2023 ON transaksi(waktu_transaksi)  
    WHERE waktu_transaksi >= '2023-01-01';
```

Kueri Uji dan Prosedur Pengukuran

Tiga kueri representatif diuji (Tabel 3). Setiap kueri dieksekusi 15 kali dalam setiap skenario. Rata-rata waktu eksekusi diukur menggunakan EXPLAIN ANALYZE. Cache database direset sebelum setiap pengujian untuk konsistensi

Tabel 3. Spesifikasi Kueri Uji Coba

ID	Tipe	Deskripsi SQL (Simplified)
Q1	Agregasi Kompleks	SELECT kategori, SUM(jumlah) FROM transaksi JOIN produk ... WHERE waktu_transaksi BETWEEN '2023-01-01' AND '2023-06-30' GROUP BY kategori;
Q2	Pencarian Spesifik	SELECT * FROM transaksi WHERE pelanggan_id = 500123 AND waktu_transaksi >= '2023-10-01';
Q3	Analisis Rentang Panjang	SELECT DATE_TRUNC('month', waktu_transaksi), COUNT(*) FROM transaksi WHERE jumlah > 100000 GROUP BY 1;

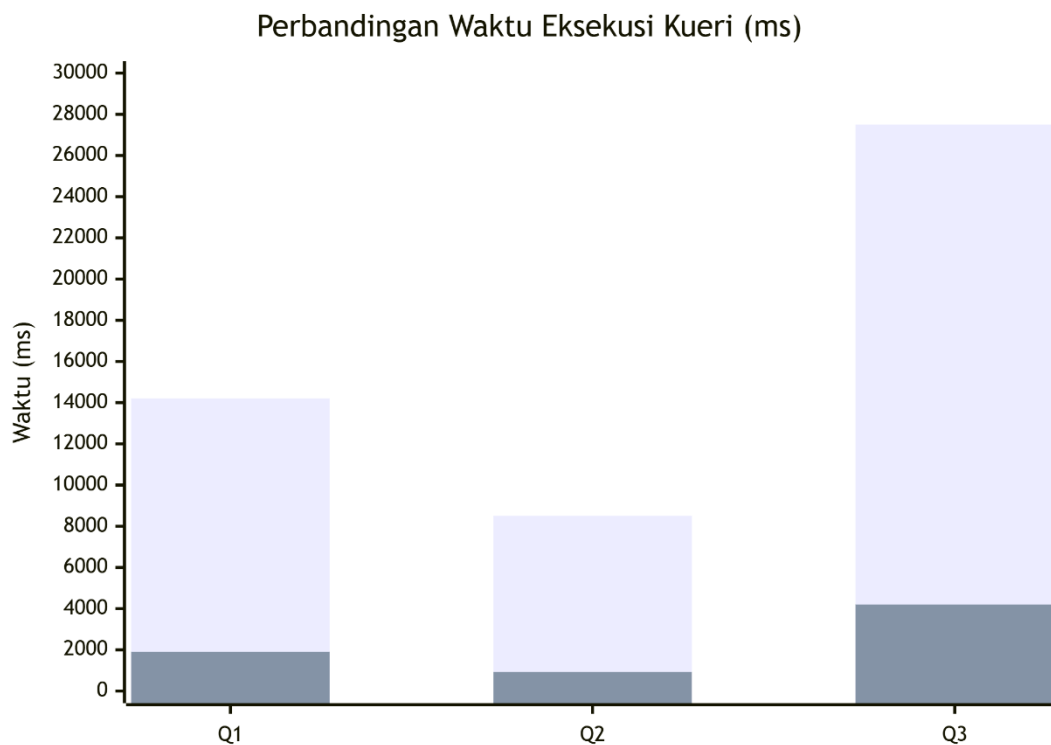
PEMBAHASAN

Hasil Pengukuran Kinerja

Hasil pengujian (Tabel 2) menunjukkan peningkatan performa yang signifikan pada Skenario B (Hybrid) untuk semua jenis kueri. Peningkatan tertinggi dicapai pada Q2 (89.1%), diikuti oleh Q1 (86.6%) dan Q3 (84.7%). Perbandingan visual ditunjukkan pada Gambar 2.

Tabel 2. Hasil Rata-Rata Waktu Eksekusi Kueri (dalam milidetik)

Kueri	Skenario A (Baseline)	Skenario B (Hybrid)	Penurunan Waktu	Peningkatan
Q1	14,200	1,900	12,300 ms	86.6%
Q2	8,500	930	7,570 ms	89.1%
Q3	27,500	4,200	23,300 ms	84.7%



Gambar 2. Grafik Perbandingan Waktu Eksekusi Kueri

Grafik batang ganda yang secara visual membandingkan waktu eksekusi (dalam ms) Skenario A dan Skenario B untuk Q1, Q2, dan Q3, menunjukkan penurunan yang drastis pada Skenario B.

Analisis Query Execution Plan

Analisis rencana eksekusi (EXPLAIN ANALYZE) mengungkap mekanisme di balik peningkatan performa:

- 1) Pada Q1 (Skenario B): Terjadi partition pruning. Optimizer hanya mengakses partisi transaksi_2023_q1 dan transaksi_2023_q2, mengabaikan 10 partisi lainnya. Ini mengurangi beban I/O dari ~100 juta baris menjadi ~16.7 juta baris.
- 2) Pada Q2 (Skenario B): Kueri melakukan Index Scan menggunakan idx_transaksi_pelanggan_id_2023 (indeks parsial) yang hanya ada pada partisi transaksi_2023_q4. Ukuran indeks yang lebih kecil mempercepat traversal.

Hasil penelitian membuktikan sinergi antara partitioning dan indexing. Partitioning berfungsi sebagai filter makro (pruning), sementara partial indexing berfungsi sebagai akselerator mikro di dalam partisi yang relevan. Strategi ini selaras sempurna dengan pola akses e-commerce di mana mayoritas kueri analitik terfokus pada data terbaru (hot data). Keuntungan strategi hybrid ini bersifat ganda: (1) Peningkatan Performa: melalui reduksi volume data yang dipindai dan efisiensi indeks; (2) Efisiensi Operasional: ukuran indeks total yang lebih kecil mengurangi penggunaan penyimpanan dan overhead maintenance. Namun, strategi ini menambah kompleksitas desain awal dan memerlukan pemantauan untuk memastikan kunci partisi tetap relevan dengan pola kueri.

KESIMPULAN

Berdasarkan hasil eksperimen, dapat disimpulkan bahwa strategi optimasi hybrid yang mengombinasikan Range Partitioning dan Partial Indexing terbukti efektif secara signifikan dalam meningkatkan performa kueri analitik pada simulasi VLDB transaksi e-commerce, dengan peningkatan waktu eksekusi berkisar antara 84,7% hingga 89,1% yang dihasilkan dari sinergi mekanisme partition pruning dan pemanfaatan indeks parsial yang lebih efisien. Temuan ini menunjukkan bahwa pendekatan tersebut mampu menjawab tantangan performa yang umum terjadi pada workload analitik e-commerce yang bersifat temporal dan selektif, sehingga direkomendasikan untuk diadopsi pada sistem e-commerce yang mengalami degradasi kinerja kueri, dengan penentuan kolom partisi dan kriteria indeks parsial yang didasarkan pada analisis pola kueri aktual. Selain itu, penelitian ini membuka peluang pengembangan lebih lanjut melalui pengujian pada lingkungan database terdistribusi, perbandingan dengan teknik optimasi lain seperti columnar storage, penggunaan dataset riil dari platform e-commerce, serta integrasi dengan mekanisme caching guna mencapai optimasi performa yang lebih komprehensif dan berkelanjutan.

DAFTAR PUSTAKA

- Lubis, N., Harahap, A. Y., Tantawi, R., Aslami, N., & Sitanggang, T. N. (2024). Dampak Perkembangan Ekonomi Digital terhadap Pertumbuhan Sektor E-Commerce di Indonesia: Perspektif Teknologi, Konsumen, dan Regulasi. *Jurnal Penelitian Ekonomi Akuntansi (JENSI)*, 8(2), 348-259. <https://doi.org/10.33059/jensi.v8i2.10649>
- Usanto S. (2023). BIG DATA IMPLEMENTATION AND USE FOR BUSINESS. *Devotion: Journal of Research and Community Service*, 4(13). Dari <https://devotion.greenvest.co.id/index.php/dev/article/download/552/996>
- Maarif, M. R. (2022). Summarizing Online Customer Review using Topic Modeling and Sentiment Analysis. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 7(3), 177-191. <https://doi.org/10.14421/jiska.2022.7.3.177-191>
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill.
- Leis, V., Boncz, P., Kemper, A., & Neumann, T. (2019). Adaptive Optimization of Very Large Join Queries. *Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19)*, 677-692. <https://doi.org/10.1145/3299869.3319890>
- PostgreSQL Global Development Group. (2023). PostgreSQL 15 Documentation. Diakses dari <https://www.postgresql.org/docs/15/index.html>
- Beyari, H. (2021). RECENT E-COMMERCE TRENDS AND LEARNINGS FOR E-COMMERCE SYSTEM DEVELOPMENT FROM A QUALITY PERSPECTIVE. *International Journal for Quality Research*, 15(3). <https://doi.org/10.24874/IJQR15.03-07>
- Cao, Z., Chu, J., Hui, K. L., & Xu, H. (2021). The relationship between online referral marketing and price promotion: Evidence from a large e-commerce platform. *Journal of Management Information Systems*, 38(3), 855-888. <https://doi.org/10.1080/07421222.2021.1962597>

Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Pearson.

Jurnal Teknik Informatika dan Sistem Informasi (JuTISI). (n.d.). Panduan Penulisan. Diakses dari https://journal.maranatha.edu/index/jutisi/panduan_penulisan

ACKNOWLEDGMENTS

Penulis mengucapkan terima kasih kepada Dosen dan Program Studi Sistem Informasi STMIK LIKMI Bandung atas dukungannya. Penelitian ini dilakukan secara mandiri tanpa sumber pendanaan eksternal.

FUNDING INFORMATION

None.

CONFLICTING INTEREST STATEMENT

The authors state that there is no conflict of interest in the publication of this article.

HISTORY OF ARTICLE

Submitted : June 30, 2025

Revised : December 16, 2025

Accepted : December 17, 2025

Published : December 17, 2025