



Modul Praktikum Sistem Manajemen Basis Data

ACID dan High Concurrency PostgreSQL

*Departemen Teknik Komputer
Institut Teknologi Sepuluh Nopember
Surabaya*

Penulis: Arta Kusuma Hernanda

2025

Bab 1

Implementasi Indexing pada PostgreSQL

1 Pendahuluan

Index pada database adalah struktur data khusus yang mempercepat operasi pencarian dan pengambilan data. Analoginya seperti indeks pada buku yang membantu kita menemukan konten tertentu dengan cepat tanpa perlu membaca seluruh buku. Dalam PostgreSQL, index disimpan dalam struktur khusus yang mengoptimalkan pencarian berdasarkan kolom tertentu.

Implementasi indexing yang tepat merupakan salah satu strategi utama untuk meningkatkan performa database, terutama pada sistem dengan data besar dan tingkat konkurensi tinggi.

1.1 Tujuan Praktikum

Dilaksanakannya praktikum indexing, praktikan diharapkan mampu:

1. Memahami konsep dasar indexing pada database PostgreSQL
2. Mempelajari tipe-tipe index di PostgreSQL
3. Mengimplementasikan index yang tepat untuk sebuah skenario
4. Menganalisis dampak index terhadap performa query konkuren

2 Konsep Dasar Indexing

2.1 Apa itu Index?

Index pada database bekerja mirip dengan indeks buku: membantu sistem menemukan data dengan cepat tanpa memeriksa seluruh tabel (table scan). Index menyimpan pointer ke baris data dalam tabel berdasarkan nilai kolom yang diindeks.

2.2 Bagaimana Index Bekerja

Ketika query dijalankan, PostgreSQL menganalisis apakah menggunakan index akan lebih efisien daripada melakukan table scan. Jika menggunakan index lebih efisien, PostgreSQL akan:

1. Mencari data di index berdasarkan kondisi WHERE
2. Menemukan referensi (pointer) ke baris data yang relevan
3. Mengambil data dari tabel menggunakan pointer tersebut

2.3 Kapan Menggunakan Index

Index sangat berguna pada:

1. Kolom yang sering digunakan dalam klausa WHERE
2. Kolom yang sering digunakan untuk JOIN antar tabel
3. Kolom yang sering digunakan dalam klausa ORDER BY atau GROUP BY
4. Tabel berukuran besar dengan query yang hanya mengambil sebagian kecil data

2.4 Kapan Tidak Menggunakan Index

Index tidak selalu bermanfaat pada:

1. Tabel kecil yang lebih efisien dilakukan sequential scan
2. Kolom dengan kardinalitas rendah (nilai yang berbeda sedikit)
3. Kolom yang jarang digunakan dalam query
4. Tabel yang sering diupdate/dihapus (overhead pemeliharaan index)

2.5 Dampak Index pada Performa

Index memberikan dampak pada:

1. **SELECT**: Mempercepat query dengan mengurangi jumlah baris yang perlu diperiksa
2. **INSERT**: Memerlukan overhead tambahan untuk memperbarui index
3. **UPDATE**: Memerlukan overhead untuk memperbarui index jika kolom yang diindex berubah
4. **DELETE**: Memerlukan overhead untuk memperbarui index

3 Jenis-jenis Index di PostgreSQL

PostgreSQL menyediakan beberapa jenis index yang dapat dipilih sesuai dengan kebutuhan:

3.1 B-tree Index

1. Index default di PostgreSQL
2. Efisien untuk perbandingan dengan operator equality (=) dan range (<, >, <=, >=, *BETWEEN*)
3. Mendukung urutan untuk ORDER BY
4. Cocok untuk kebanyakan kasus penggunaan

3.2 Hash Index

1. Optimal hanya untuk operator equality (=)
2. Lebih cepat dari B-tree untuk operasi equality sederhana
3. Tidak mendukung range queries atau sorting
4. Dirancang untuk tabel hash di memori

3.3 GiST Index (Generalized Search Tree)

1. Index untuk data geometri, text-search, dan data kompleks lainnya
2. Fleksibel, dapat digunakan untuk data custom
3. Mendukung nearest-neighbor searches
4. Digunakan untuk full-text search, data geografis, dll

3.4 GIN Index (Generalized Inverted Index)

1. Dirancang untuk nilai yang memiliki multiple components (arrays, jsonb, text-search)
2. Efisien untuk pencarian yang membutuhkan matching multiple values
3. Cocok untuk kolom yang menyimpan data semi-structured
4. Lebih lambat untuk operasi insert dibanding GiST

3.5 BRIN Index (Block Range INdex)

1. Dirancang untuk tabel sangat besar dengan data yang terurut secara natural
2. Sangat kecil dan efisien untuk kolom seperti timestamp, ID sequential, dll
3. Kinerja query lebih rendah dari B-tree tapi overhead penyimpanan jauh lebih kecil

3.6 SP-GiST Index (Space-Partitioned GiST)

1. Untuk data yang bisa dipartisi secara non-overlapping
2. Baik untuk data hierarchical seperti rentang IP, geo-data
3. Mendukung nearest-neighbor searches

Jenis Index	Karakteristik dan Penggunaan
B-tree	Index default PostgreSQL. Efisien untuk operasi perbandingan (=, <, >, BETWEEN) dan mendukung pengurutan (ORDER BY).
Hash	Khusus untuk operasi equality (=). Lebih cepat dari B-tree untuk lookup sederhana, tidak mendukung range.
GiST/GIN	Untuk data kompleks seperti full-text search, data spasial, array, dan JSON. GIN lebih lambat untuk insert tapi lebih cepat untuk search.
BRIN	Untuk tabel besar dengan data terurut secara natural (seperti timestamp). Ukuran kecil, overhead rendah.

Gambar 1.1: Jenis-jenis Index di PostgreSQL dan Penggunaannya

3.7 Benchmark pada Konkurensi Tinggi

Pada situasi konkurensi tinggi, performa index dapat dipengaruhi oleh beberapa faktor:

- **Contention:** Kompetisi antar koneksi untuk mengakses data yang sama
- **Lock contention:** Waktu tunggu karena lock pada baris atau tabel
- **Index bloat:** Overhead karena index yang jarang di-maintenance

4 Best Practices Implementasi Indexing

1. Index kolom yang sering digunakan dalam WHERE, JOIN, ORDER BY
2. **Hindari over-indexing:** Terlalu banyak index berdampak negatif pada performa INSERT/UPDATE/DELETE
3. **Gunakan composite index** untuk query dengan multiple conditions
4. **Perhatikan urutan kolom** pada composite index (most selective first)
5. **Pertimbangkan partial index** untuk subset data yang sering diakses
6. **Gunakan EXPLAIN ANALYZE** untuk validasi penggunaan index

7. **Jalankan VACUUM ANALYZE** secara berkala untuk memperbarui statistik
8. **Monitor ukuran dan penggunaan index** menggunakan `pg_stat_*`
9. **Lakukan REINDEX** pada index yang terfragmentasi
10. **Evaluasi trade-off** antara kecepatan query vs overhead pemeliharaan

5 Referensi

- [PostgreSQL Documentation: Indexes](#)
- [PostgreSQL Documentation: EXPLAIN](#)
- [PostgreSQL Documentation: Performance Tips](#)
- [PostgreSQL Documentation: Index Types](#)
- [PostgreSQL Documentation: pgbench](#)