Bab 4

Optimasi Query

POKOK BAHASAN:

- ✓ Optimasi Perintah SQL
- ✓ Informasi Jalur Akses Query
- ✓ Faktor-faktor yang berpengaruh terhadap kecepatan akses data

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami latar belakang diperlukannya optimasi query
- ✓ Memahami cara melakukan optimasi perintah SQL
- ✓ Memahami faktor-faktor yang berpengaruh terhadap kecepatan akses data

4.1. PENDAHULUAN

Data yang tersimpan dalam database semakin lama akan semakin besar ukuran atau volumenya. Kalau tidak didukung dengan kecepatan akses yang memadai maka akan semakin menurun unjuk kerjanya. Ukuran unjuk kerja dalam hal ini kecepatan akses data dipengaruhi oleh banyak faktor. Pada bab ini akan membahas tentang optimasi query serta faktor-faktor lain yang berpengaruh terhadap optimalisasi kecepatan akses data.

4.2. OPTIMASI PADA PERINTAH SQL

Desain aplikasi saja tidak cukup untuk meningkatkan unjuk kerja harus didukung dengan optimasi dari perintah SQL yang digunakan pada aplikasi tersebut. Dalam mendesain database, seringkali lokasi fisik data tidak menjadi perhatian penting. Karena hanya desain logik saja yang diperhatikan. Padahal untuk menampilkan hasil query dibutuhkan pencarian yang melibatkan struktur fisik penyimpanan data. Inti dari optimasi query adalah meminimalkan "jalur" pencarian untuk menemukan data yang disimpan dalam lokasi fisik.

Index pada database digunakan untuk meningkatkan kecepatan akses data. Pada saat query dijalankan, index mencari data dan menentukan nilai ROWID yang membantu menemukan lokasi data secara fisik di disk. Akan tetapi penggunaan index yang tidak tepat, tidak akan meningkatkan unjuk kerja dalam hal ini kecepatan akses data.

Misal digunakan index yang melibatkan tiga buah kolom yang mengurutkan kolom menurut kota, propinsi dan kode pos dari tabel karyawan, sebagai berikut :

```
CREATE INDEX idx_kota_prop_kodepos
ON karyawan(kota, propinsi, kode_pos)
TABLESPACE INDX;
```

Kemudian user melakukan query sebagai berikut :

```
SELECT * FROM karyawan WHERE propinsi='Jawa Barat';
```

Pada saat melakukan query ini, index tidak akan digunakan karena kolom pertama (kota) tidak digunakan dalam klausa WHERE. Jika user sering melakukan query ini, maka kolom index harus diurutkan menurut propinsi. Selain itu, proses pencarian data akan lebih cepat jika data terletak pada block tabel yang berdekatan daripada harus mencari di beberapa datafile yang terletak pada block yang berbeda.

Misal pada perintah SQL berikut ini :

```
SELECT * FROM karyawan
WHERE id BETWEEN 1010 AND 2010;
```

Query ini akan melakukan "scan" terhadap sedikit data block jika tabel karyawan diatas diurutkan berdasarkan kolom id. Untuk mengurutkan berdasarkan kolom yang berbeda-beda maka tabel disimpan dalam flat file, kemudian tabel diekspor dan diurutkan sesuai kebutuhan.

Alternatif yang lain, bisa digunakan perintah untuk membuat tabel lain yang memiliki urutan yang berbeda dari tabel asal, seperti perintah SQL berikut :

```
CREATE TABLE karyawan_urut
AS SELECT * FROM karyawan
ORDER BY id;
```

Pada SQL diatas, tabel karyawan_urut berisi data yang sama dengan tabel karyawan hanya datanya terurut berdasarkan kolom id.

4.3. PERENCANAAN EKSEKUSI

Bagaimana cara melihat jalur akses yang akan digunakan database saat melakukan query ? Pada Database Oracle, informasi ini dapat dilihat dengan menggunakan perintah explain plan, yang akan memberi informasi tentang rencana eksekusi dari suatu query. Informasi ini disimpan dalam tabel PLAN_TABLE yang terdapat di schema user yang mengeksekusi perintah tersebut.

Sebelum melakukan perintah explain plan, terlebih dahulu buat table PLAN_TABLE dengan menggunakan script utlxplan.sql yang diambil dari \%ORACLE_HOME%\RDBMS\ADMIN.

Setelah itu table PLAN_TABLE dapat digunakan seperti contoh berikut :

```
SQL> explain plan
Set statement_id='test1'
Into plan_table for
Select * from karyawan where gaji=2000000;
```

Dalam PLAN_TABLE rencana eksekusi diatas dikenal dengan nama test1 yang terdefinisi pada kolom statement_id.

Untuk melihat rencana eksekusi dari test1, digunakan perintah SELECT berikut :

```
SELECT LPAD(' ',2*Level)||Operation||' '||Options||' '||Object_Name Q_Plan
FROM plan_table
WHERE statement_id='test1'
CONNECT BY PRIOR id=parent_id AND statement_id='test1'
START WITH id=0 AND statement_id='test1';
```

Contoh hasil dari eksekusi query tersebut :

```
Q_PLAN
----------------------------------------------------------------
SELECT STATEMENT
        TABLE ACCESS FULL KARYAWAN
```

Output tersebut dibaca mulai dari yang indent-nya paling dalam yaitu : TABLE ACCESS FULL KARYAWAN. Dikarenakan klausa WHERE melibatkan kolom gaji namun kolom gaji tidak ada index-nya, maka Oracle melakukan full table scan. Setelah seluruh tabel karyawan selesai dibaca, selanjutnya adalah SELECT STATEMENT yang berfungsi untuk menampilkan hasil query.

4.4. FAKTOR LAIN YANG BERPENGARUH TERHADAP KECEPATAN AKSES DATA

Faktor lain yang berpengaruh terhadap kecepatan akses data, tidak hanya terletak pada optimasi perintah SQL, tapi terhadap hal-hal lain yang berpengaruh. Diantaranya adalah optimasi aplikasi dan penggunaan cluster dan index. Hal yang akan dibahas dalam optimasi query berikut ini tidak melibatkan penggunaan komponen yang ada dalam Arsitektur *database engine,* misal pada database Oracle kecepatan akses data dipengaruhi oleh penyesuaian pada shared pool, buffer cache, redo log buffer dan sistem operasi yang digunakan.

4.4.1. OPTIMASI APLIKASI

Dalam pembuatan aplikasi, yang perlu mendapat perhatian adalah apakah akses terhadap data sudah efisien. Efisien dalam hal penggunaan obyek yang mendukung kecepatan akses, seperti index atau cluster. Kemudian juga bagaimana cara database di-desain. Apakah desain database sudah melakukan normalisasi data secara tepat.

Kadangkala normalisasi sampai level yang kesekian, tidak menjamin suatu desain yang efisien. Untuk membuat desain yang lebih tepat, kadang setelah melakukan normalisasi perlu dilakukan denormalisasi. Misalnya tabel yang hubungannya one-to-one dan sering diakses bersama lebih baik disatukan dalam satu tabel.

4.4.2. CLUSTER DAN INDEX

Cluster adalah suatu segment yang menyimpan data dari tabel yang berbeda dalam suatu struktur fisik disk yang berdekatan. Konfigurasi ini bermanfaat untuk akses data dari beberapa tabel yang sering di-query. Penggunaan cluster secara tepat dilaksanakan setelah menganalisa tabel-tabel mana saja yang sering di-query secara bersamaan menggunaan perintah SQL join.

Jika aplikasi sering melakukan query dengan menggunakan suatu kolom yang berada pada klausa WHERE, maka harus digunakan index yang melibatkan kolom tersebut. Penggunaan index yang tepat bergantung pada jenis nilai yang terdapat dalam kolom yang akan diindex. Dalam RDBMS Oracle, index B-Tree digunakan untuk kolom yang mengandung nilai yang cukup bervariasi, sedangkan untuk nilai yang tidak memiliki variasi cukup banyak, lebih baik menggunakan index bitmap.

RINGKASAN:

- Data yang tersimpan dalam jumlah yang sangat besar, Terdapat aturan system informasi dalam organisasi, system basis data dilihat sebagai bagian system informasi dalam aplikasi berskala besar.
- Untuk meningkatkan unjuk kerja tidak hanya desain logik saja yang diperhatikan tapi juga struktur fisik penyimpanan data.
- Penggunaan Index pada database secara tepat, dapat digunakan untuk meningkatkan kecepatan akses data.
- Informasi tentang jalur akses yang digunakan oleh database untuk melaksanakan query dalam database Oracle dapat dengan menggunakan perintah *explain plan*.
- Selain optimasi perintah SQL, faktor lain yang berpengaruh terhadap kecepatan akses data adalah optimasi aplikasi dan penggunaan cluster dan index.
- Pada sebuah *database engine* semisal pada database Oracle kecepatan akses data dipengaruhi oleh beberapa komponen arsitektur pembentuknya seperti shared pool, buffer cache, dan redo log buffer.
- Optimasi aplikasi tergantung pada efisiensi penggunaan obyek yang mendukung kecepatan akses seperti index atau cluster, dan normalisasi data pada desain database.

LATIHAN SOAL :

1. Apa latar belakang dari diperlukannya optimalisasi kecepatan akses data ?
2. Optimasi query dalam hubungannya dengan desain database melibatkan dua hal, yaitu ….. dan ……
3. Proses pencarian data yang telah diindeks akan lebih cepat jika data yang dicari terletak pada ……
4. Bagaimana cara melihat jalur akses yang akan digunakan database saat melakukan query ? Tunjukkan tahap-tahap yang digunakan untuk melakukan hal tersebut !
5. Sebutkan factor-faktor lain yang berpengaruh terhadap kecepatan akses data selain optimasi pada perintah SQL !
6. Beberapa *database engine* melibatkan komponen pada arsitekturnya untuk disesuaikan agar akses data lebih cepat dan efisien, berikan contohnya !