

Materi VII

Nama : Roy Agus Martin Marbun

Kelas : XII – RPL

Mata Pelajaran : Basis Data

BAB II

Perintah SQL dan Basis Data pada RDBMS

A. Arsitektur Sistem RDBMS (*Relational Database Management System*)

Sebuah *database* dibangun dengan banyak tabel, di mana tabel tersebut terdiri atas kolom yang dilengkapi dengan baris yang bertindak sebagai data atau konten. Namun, dalam mengolah suatu database terintegrasi diperlukan suatu konsep untuk menghubungkan tabel yang satu dengan yang lainnya melalui kunci (*key*) yang dimiliki. Hal itulah yang disebut dengan RDBMS (*Relational Database Management System*).

Sebuah sistem manajemen basis data relasional atau dalam bahasa Inggrisnya dikenal sebagai *relational database management system* (RDBMS) adalah sebuah program komputer (atau secara tipikal adalah seperangkat program komputer) yang didesain untuk mengatur/memanajemen sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur dan melakukan operasi-operasi data atas permintaan penggunaannya.

Contoh penggunaan DBMS ada banyak sekali dan dalam berbagai bidang kerja, misalnya akuntansi, manajemen sumber daya manusia dan lain sebagainya. Meskipun pada awalnya DBMS hanya dimiliki oleh perusahaan-perusahaan berskala besar yang memiliki perangkat komputer yang sesuai dengan spesifikasi standar yang dibutuhkan untuk mendukung jumlah data yang besar.

1. Arsitektur RDBMS

Arsitektur RDBMS memiliki banyak karakteristik yang membedakan dari model penyimpanan data lainnya. Perbedaan yang paling penting adalah pemisahan segi fisik dari segi logika suatu data. Dalam RDBMS, seluruh data secara logika tersimpan di dalam tabel-tabel, yang merupakan kumpulan dari baris dan kolom.

Sistem pencarian data di dalam RDBMS menggunakan index yang merupakan struktur data yang terpisah dari tabel dan hanya menyimpan nilai yang terstruktur dari kolom-kolom dan alamat fisiknya. Disamping itu dengan didukung oleh penggunaan *index* dapat mempercepat proses pencarian data di dalam *database*.

Faktor penting lainnya dari arsitektur RDBMS adalah *integrity constraints*. Dengan *Integrity Constraints* tabel-tabel dihubungkan dengan *key*. *Key* adalah beberapa kolom atau kombinasi kolom-kolom yang secara unik mengidentifikasi setiap tabel. Sebuah *key* yang secara unik bagi suatu tabel dapat berdiri sebagai kolom yang tidak unik bagi tabel lainnya. *Integrity Constraints* adalah aturan “*build in*” yang secara otomatis berpengaruh dalam mempertahankan integritas *database*.

Aturan-aturan integritas ini biasanya dibuat atau dirancang oleh seorang perancang *database*. Karakteristik penting lainnya dari arsitektur RDBMS adalah adanya “*Optimizer*”. *Optimizer* adalah sebuah sistem pakar yang bertugas untuk menentukan cara pemrosesan yang paling efisien bagi suatu *database*.

2. Elemen-elemen RDBMS

Adapun elemen-elemen RDBMS, diantaranya sebagai berikut.

- a. *Database*, yaitu sekelompok tabel data berisi informasi yang berhubungan. Perhatikan bahwa suatu *database* dapat terdiri dari satu tabel saja.
- b. Tabel, yaitu sekelompok *record* data, masing-masing informasi yang sejenis. Misalnya Katalog Perpustakaan, katalog itu sendiri merupakan tabel data.
- c. *Record*, yaitu entri tunggal dalam tabel yang terdiri dari sejumlah field data. Jika dalam katalog tadi, *record* adalah salah satu baris dari entri tunggal.
- d. *Field*, yaitu item tertentu dari data dalam *record*. Dalam buku telepon, sekurang-kurangnya dapat dikenali empat *field* yaitu nama keluarga, nama depan, alamat dan nomor telepon.
- e. *Index*, yaitu tipe tabel tertentu yang berisi nilai-nilai *field* kunci atau *field* yang sudah ditetapkan oleh pemakai ke lokasi *record* yang sebenarnya. Nilai-nilai dan pointer ini disimpan dalam urutan tertentu dan mungkin digunakan untuk menyajikan data dalam urutan *database*.
- f. *Query*, yaitu perintah SQL yang dirancang untuk memanggil kelompok *record* tertentu dari satu tabel atau lebih untuk melakukan operasi pada tabel. Meskipun perintah SQL dapat dijalankan langsung dari program, *query* sebagai mesin perintah dapat menyimpannya sendiri kedalam *database*.
- g. *Filter*, sebenarnya bukan merupakan bagian dari *database*, namun ia digunakan bersama urutan *index* dan pengurutan untuk menentukan data mana yang diproses atau ditampilkan. *Filter* merupakan pembatas kondisi yang dikenakan kepada data.
- h. *View*, terdiri atas jumlah *record* yang tampak (atau diproses) dan urutan penampilannya dan pemrosesannya. *Filter* dan *index* adalah yang dapat mengendalikan *View* secara khusus.