**BAB 14 Distributed Database**

**Basis Data Terdistribusi**

Basis data yg tersebar di beberapa lokasi yang tiap lokasinya dikelola oleh DBMS yang mandiri dan terhubung melalui jaringan.

**Sistem Komputasi Terdistribusi**

Kumpulan proses yg bekerja sama dalam melakukan suatu tugas yg terhubung melalui jaringan.

**DBMS**

Software pengelolaan basis data terdistribusi.

**Tujuan Distributed Database**

Memudahkan pengguna mengakses data dari beberapa tempat

**Karakteristik Distributed Database**

1. Disimpan di lokasi yg berbeda
2. Tiap lokasi database dihubungkan dengan jaringan komputer
3. Tiap database server memiliki DBMS nya masing-masing

**Taksonomi Distributed Database**

1. **Homogenous**

Distributed database yg DBMS nya sama atau sejenis.

**- Autonomus**

Tiap server memiliki DBMS yang independen dan dapat mengelola data secara mandiri. mudah untuk dikelola

**- Non Autonomus**

Seluruh pengelolaan harus terkoordinasi dengan DBMS pusat. semua schema lokal akan disatukan menjadi skema global atau.

1. **Heterogenous**

Distributede database yg DBMS nya berbeda beda.

**- System**

Tiap server memiliki kemampuan functional yg berbeda ada yg penuh dan parsial

**- Full DBMS Functionality**

Functional penuh pada DBMS per server

**- Partial Multidatabase**

Functionalnya beberapa aja

**- Federated**

DB lokal yg melayani data unik yg hanya dibutuhkan di beberapa tempat tertentu. Tipenya :

**- Loose Integration**

Database lokal melayani data unik dan skemanya beda-beda

**- Thight Integration**

Database lokal melayani data unik dan pakai skema global/umum

**- Unfederated**

Semua akses data harus melalui pusat koordinasi data

**- Gateways**

Model jalur interkoneksi sederhana antar database tapi tidak pakai database logis. kalau diakses di lokal ya pake skema lokal, kalau di akses di global/pusat/remote ya skema global

**Pemahaman Umum Distributed Database**

1. **Transparansi Lokal**

- User tidak harus tau lokasi data pada server mana. tetapi data akan tetap bisa diambil dengan cepat

- Tiap request data akan diolah oleh distributed db sehingga akan diarahkan ke server dimana data tsb disimpan

1. **Otonomi Lokal**

- server lokal akan tetap jalan walau ada masalah jaringan

- setiap server mengontrol datanya sendiri, jadi dalam bbrp fungsi server punya wewenang penuh

**Alasan Memakai Distributed Database**

- Jika butuh otonomi dan desentralisasi unit-unit usaha atau wewenang pada unit usaha

- Jika butuh sharing data

- Jika ingin meningkatkan keandalan dan ketepatan sistem

- Jika ada app dari vendor yg berbeda

- Jika ingin recovery data

**Keuntungan**

- Sharing data

- Available dan Reliability, akan ttp jalan walau salah satunya mati

- Kecepatan Proses Query

- Otonomi lokal

- Efisien & fleksibel

**Kerugian**

- biaya tinggi

- resiko problem yg lebih besar

- biaya proses tinggi

**Arsitektur Distributed System**

Ada 3 artsitektur yg perbedaan nya di DBMS, yaitu bagian client-server, kolaborasi server dan middleware.

1. **Sistem client-server**

client menangani dengan UI sedangkan server mengelola data dan transaksi.

- relatif mudah dan populer

- beban kerja server cukup tinggi

- biaya tinggi karena proses tinggi

- perlu caching

1. **Sistem kolaborasi server**

- Query yg dikirim akan di pecah menjadi sub query dan disebar ke beberapa server sesuai kebutuhan.

- Perlu memperhatikan biaya jaringan dan pengolahan lokal dalam dekomposisi query

1. **Sistem middleware**

- membuat satu query dapat dijalankan di multiple server tanpa diekseskusi di multi place

- untuk dekomposisi query dan join dilakukan oleh software yg disebut middleware

**Teknik Pengaturan Data**

1. **Syncronous**

- data yg terhubung pada jaringan akan selalu terupdate

- semua duplikat data selalu identic atau strukturnya sama

- perubahan data berlaku ke semua duplikasi data

- integritas tinggi

- biaya overhead tinggi

1. **Asyncronous**

- menyediakan replikasi data pada bbrp node sehingga local server dapat mengakses data tanpa harus keluar jaringan local

- inkonsistensi data temporer mungkin terjadi

- perubahan data tidak segera di ubah di replikasi

- integritas data agak rendah

- biaya overhead lebih rendah a.k.a response time faster

**Pendistribusian Basis Data**

1. **Replikasi Data**

Duplikasi data disebar ke beberapa lokasi yg berbeda pada server

1. **Partisi Horizontal**

Duplikasi dilakukan terhadap record data

1. **Partisi Vertikal**

Duplikasi dilakukan pada column data

**DBMS Terdistribusi**

DBMS buat distributed database.

- Distributed data dictionary : membantu untuk mencari data di sebaran lokasi

- Menentukan lokasi pengambilan data by query

- Middleware : translasi dan koordinasi antar DBMS yg beda

- Multiphase Commit protocol : menjaga konsistensi data

- Pengendalian primary key global