

Database dalam Sistem Terdistribusi

Modern Database Management

6th Edition

*Jeffrey A. Hoffer, Mary B. Prescott, Fred R.
McFadden*

Definisi

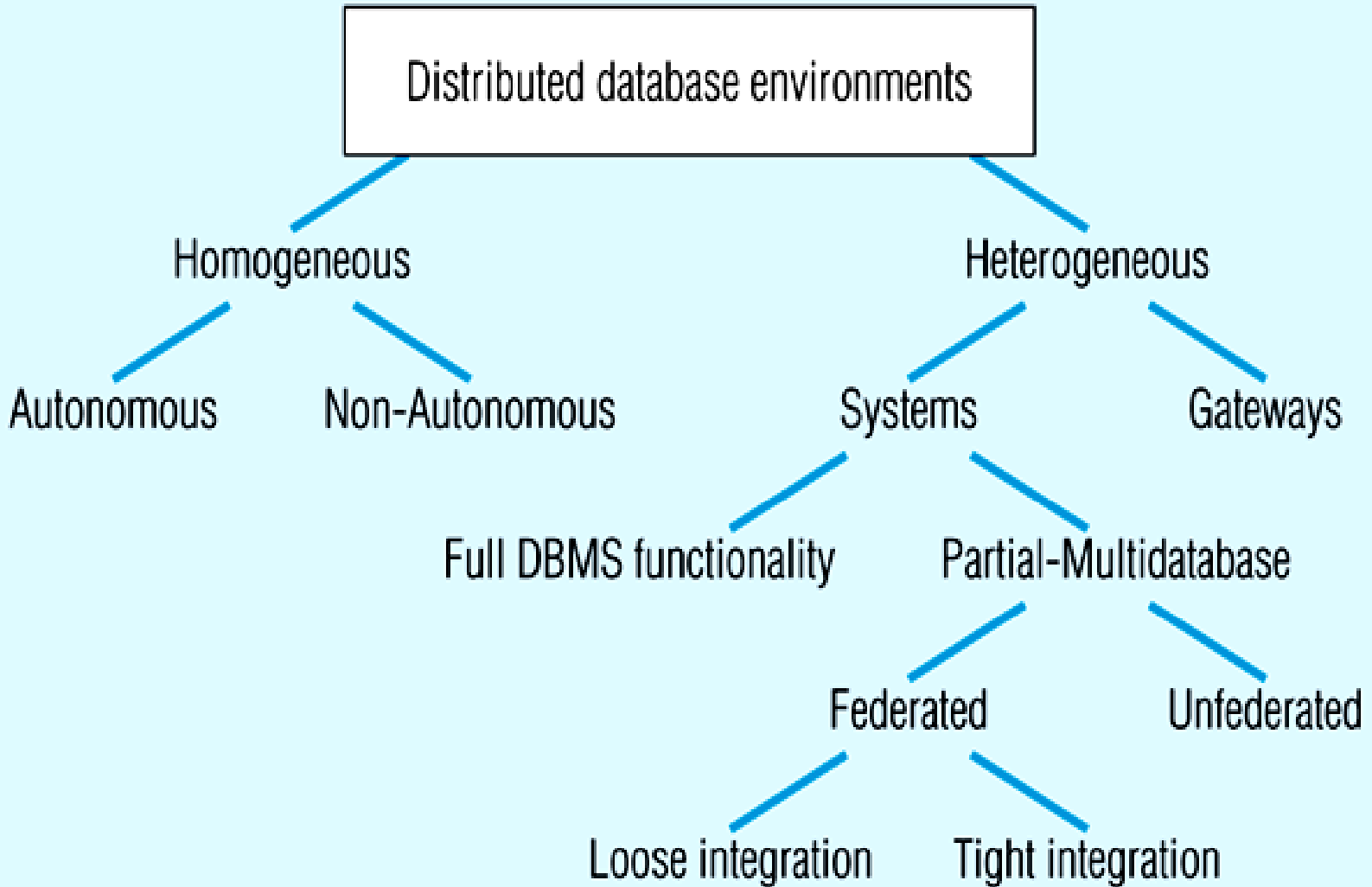
- **Database terdistribusi:** *Sebuah database logic yang tersebar secara fisik* diantara komputer-komputer yang berada di lokasi berbeda yang dihubungkan dengan perangkat komunikasi
- **Desentralisasi Database:** Sekumpulan database *mandiri* dalam komputer-komputer yang tidak saling terhubung

Keduanya TIDAK sama !

Alasan penggunaan Database Terdistribusi

- Otonomi dari unit bisnis yang letaknya berjauhan
- Pertukaran data
- Biaya komunikasi data
- Kehandalan komunikasi data dan biaya
- Keragaman sistem aplikasi
- Pemutakhiran database
- Proses analisa dan transaksi

Lingkungan database terdistribusi (*Bell and Grimson, 1992*)



Pilihan database terdistribusi

- Homogen – DBMS yang sama di setiap node
 - DBMS yang otonom dan mandiri
 - DBMS yang tidak otonom – terpusat, pengkoordinir
 - Mudah dalam pengelolaannya, sulit dalam mengendalikan
- Heterogen – Setiap node mempunyai DBMS sendiri
 - Sistem – dengan peran DBMS yang penuh atau sebagian
 - Gateway – jalur yang sederhana antar database tidak mengutamakan salah satu database logic
 - Sulit mengelolanya, tergantung pada kemandirian organisasi

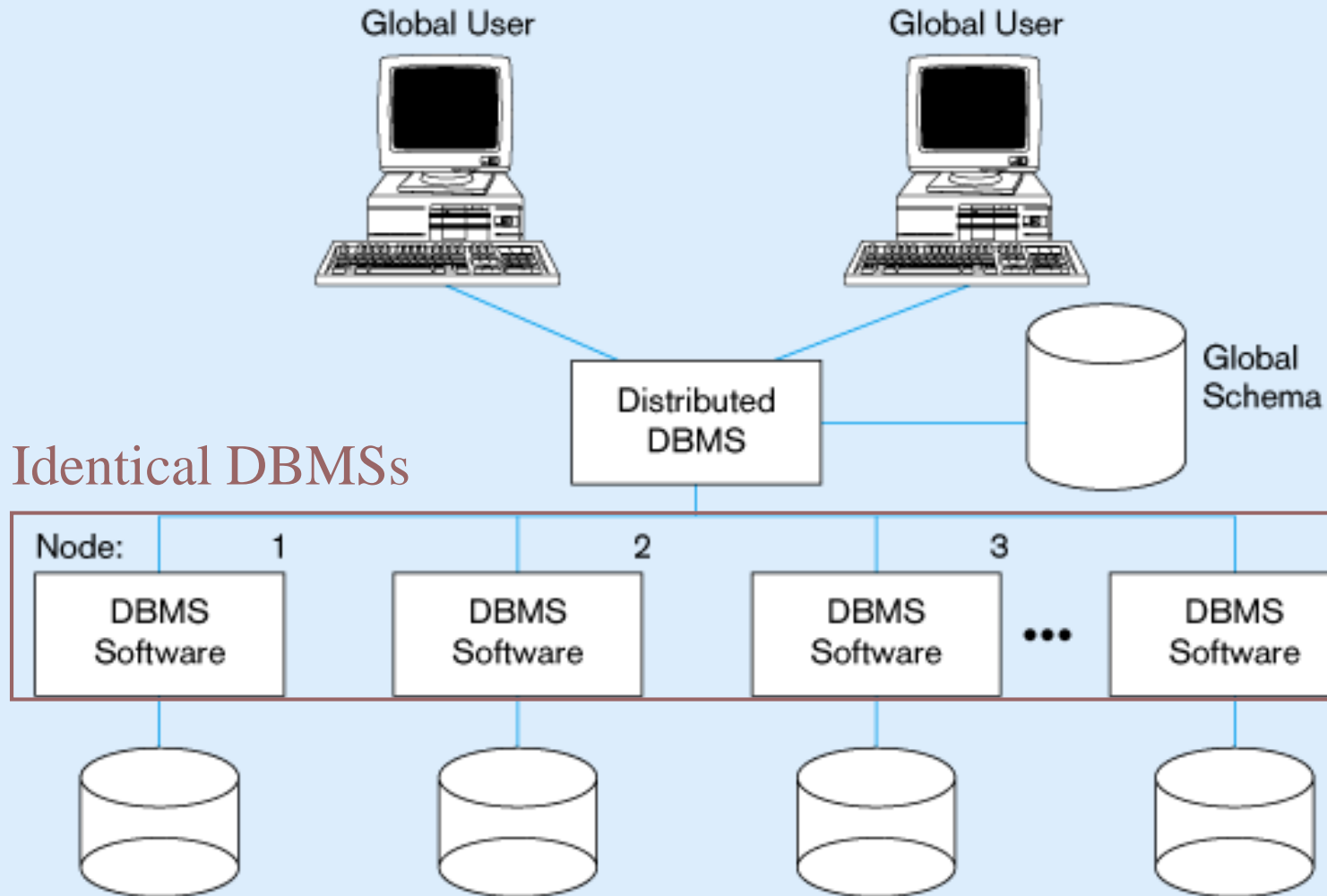
Pilihan database terdistribusi

- **Sistem** – mendukung semua atau beberapa fungsi database logic
- Fungsi DBMS penuh – Semua fungsi DB tersebar
 - Sebagian – Banyak – beberapa fungsi DB tersebar
 - Federasi – Mendukung db lokal untuk permintaan data yang unik
 - Kehilangan keterpaduan – DB lokal memiliki skema tersendiri
 - Keterpaduan tinggi – DB lokal menggunakan skema umum
 - Unfederated – Memerlukan semua akses ke pusat, untuk koordinasi modul

DB Homogen, tanpa otonomi

- Data didistribusikan di setiap node
- DBMS seragam untuk setiap node
- Semua data dikelola oleh DBMS yang terdistribusi (bukan data lokal yang eksklusif)
- Semua akses seragam dengan skema yang global
- Skema global adalah gabungan dari semua skema lokal

Database Homogen

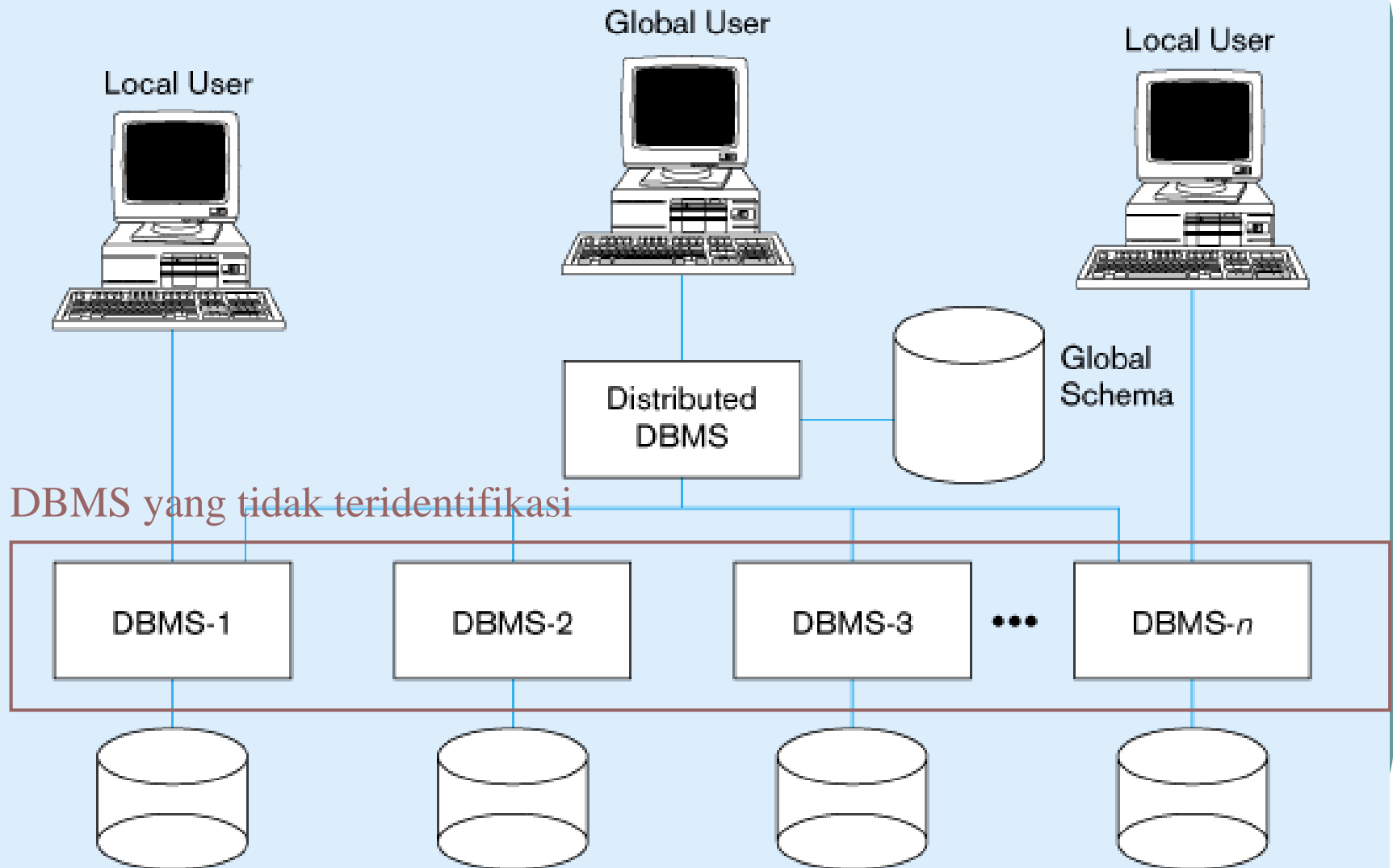


Source: adapted from Bell and Grimson, 1992.

Lingkungan heterogen yang khusus

- Data tersebar di setiap node
- DBMS yang berlainan mungkin digunakan di setiap node
- Akses lokal menggunakan DBMS dan skema lokal
- Akses jarak jauh menggunakan skema global

Lingkungan Heterogen Khusus



Source: adapted from Bell and Grimson, 1992.

Tujuan Umum

- **Transparansi Lokasi**
 - User tidak mengetahui lokasi data
 - Data yang dibutuhkan otomatis diteruskan ke tempat yang tepat
- **Otonomi Lokal**
 - Situs Lokal dapat mengoperasikan databasenya ketika koneksi dengan jaringan terputus
 - Setiap situs dapat mengendalikan data, keamanan, login dan pemutakhirannya

Lain-lain :

- Database terdistribusi sinkron
 - Semua kopi data selalu teridentifikasi
 - Data terkini secara cepat diterapkan ke seluruh kopian melalui jaringan
 - Baik untuk integritas data
 - Biaya tinggi → waktu respon lambat
- Database Terdistribusi Asinkron
 - Mentolerir data yang tidak konsisten
 - Perkembangan data terkini tertunda
 - Integritas data rendah
 - Biaya murah → waktu respon cepat

Keuntungan database terdistribusi menggunakan database terpusat

- Meningkatkan kepercayaan
- Pengendalian lokal terhadap data
- Pertumbuhan modul
- Biaya komunikasi rendah
- Respon cepat untuk queri tertentu

Kerugian database terdistribusi dibandingkan database terpusat

- Biaya software dan rumit
- Biaya proses
- Integritas data terbuka
- Respon lambat untuk queri tertentu

Beberapa pilihan untuk mendistribusikan database

- Replikasi Data
 - Kopi data didistribusikan ke situs berbeda
- Partisi Horizontal
 - Baris lain dalam sebuah tabel didistribusikan ke situs yang berbeda
- Partisi Vertikal
 - Kolom lain dalam sebuah tabel didistribusikan ke situs berbeda
- Kombinasi ketiganya

Replikasi Data

- Keuntungan -
 - Dapat dipercaya
 - Respon cepat
 - Dapat menghindari dampak integritas transaksi terdistribusi (jika replikasi data diperbarui secara berkala)
 - Memasangkan kembali node (transaksi selesai dengan lengkap jika beberapa node mati)
 - Mengurangi lalulintas jaringan pada waktu sibuk (jika update data dapat di tunda)

Replikasi Data

- Kerugian -
 - Membutuhkan tambahan penyimpanan data
 - Membutuhkan waktu tambahan untuk operasi update
 - Kompleksitas dan biaya update
 - Integritas yang terbuka terhadap data yang tidak benar jika replikasi tidak diupdate terus menerus

Oleh karena itu, lebih baik digunakan untuk data yang read-only

Jenis replikasi

- **Push Replication –**
 - Situs yang diupdate mengirim perubahan ke situs yang lain
- **Pull Replication –**
 - Situs penerima yang mengatur ketika pesan untuk mengupdate akan diproses

Jenis Push Replication

- Snapshot Replication -
 - Secara periodik mengirimkan perubahan ke situs master
 - Master mengumpulkan updating dalam log
 - Perubahan perbagian atau keseluruhan
 - Dinamis vs. pembagian hak update
- Near Real-Time Replication -
 - Penyebaran update tanpa memerlukan konfirmasi
 - Selesai penggunaan sebagai pemicu
 - Pesan update disimpan di antrian pesan sampai diproses situs penerima

Lain-lain tentang Replikasi Data

- Data tepat waktu – toleransi yang tinggi untuk data yang kadaluwarsa mungkin dibutuhkan
- Kemampuan DBMS – jika DBMS tidak dapat mendukung query dengan multi-node, replikasi sangat dibutuhkan
- Dampak pada kinerja – pemutakhiran mungkin akan menyebabkan masalah pada kinerja sebuah node yang sibuk
- Keragaman jaringan – replikasi yang rumit dan kompleks
- Kemampuan komunikasi jaringan – Pemutakhiran akan memberi beban berat pada komunikasi

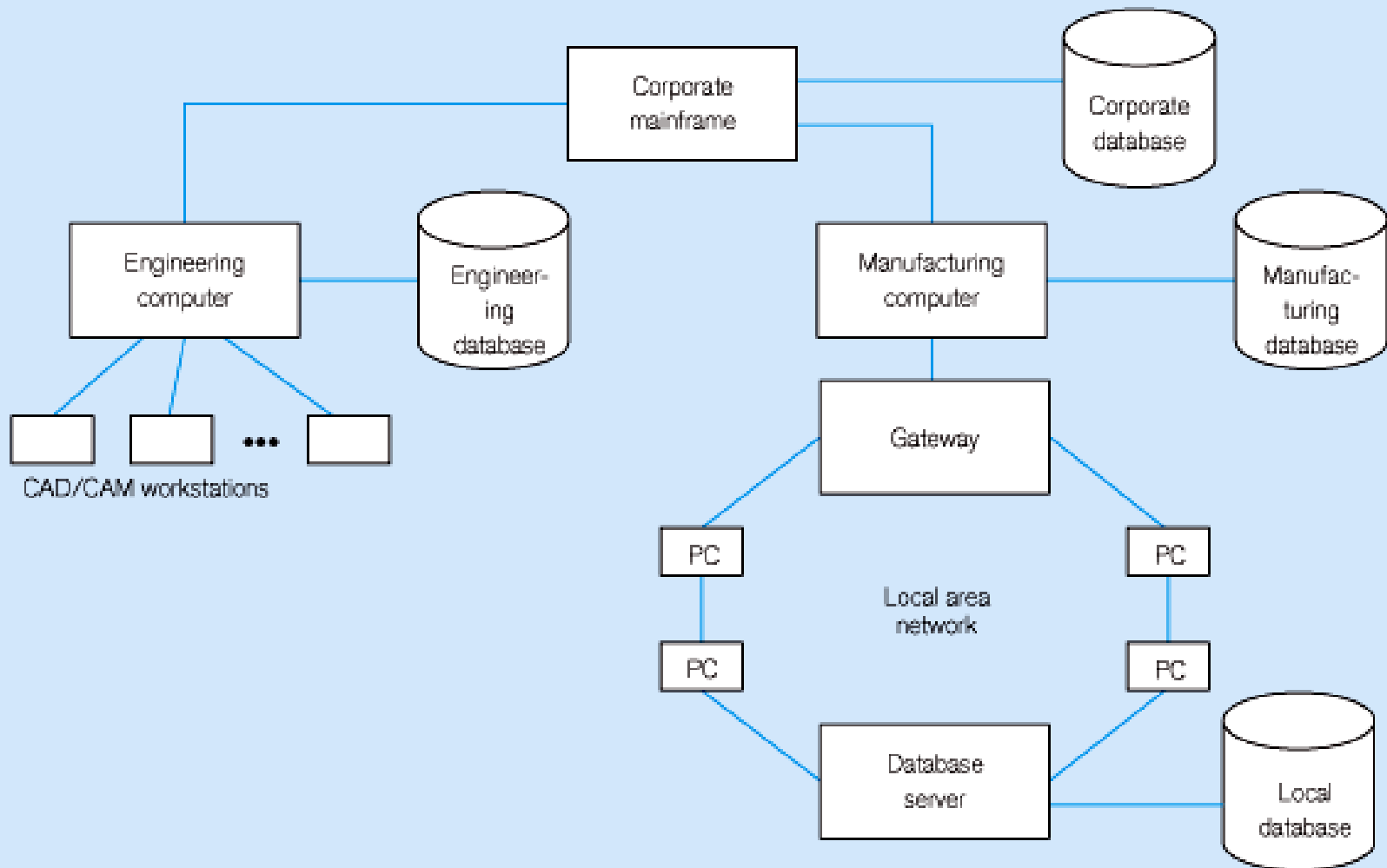
Partisi Horizontal

- Sebagian baris sebuah tabel berada di situs yang lain
- Keuntungan -
 - Simpanan data tertutup pada saat digunakan → efisien
 - Akses lokal optimal → kinerja lebih baik
 - Hanya daya yang sesuai yang tersedia → aman
 - Gabungan antar partisi → mudah dalam query
- Kerugian
 - Akses data antar partisi → kecepatan proses tidak konsisten
 - Tidak ada data replikasi → backup rawan

Partisi Vertikal

- Sebagian kolom dari sebuah tabel berada di situs lain
- Keuntungan dan kerugian sama dengan pada partisi horisontal kecuali kombinasi data antar partisi lebih sulit karena memerlukan proses join (sebagai pengganti union)

Sistem Proses Terdistribusi sebuah perusahaan pabrikan



Lima pengelompokkan database tersebar

- ★ Database terpusat, akses tersebar
- ★ Replikasi dengan update permukaan secara periodik
- ★ Replikasi dengan update sinkron yang hampir *real time*
- ✧ Pemartisian, satu database logic
- ✧ Pemartisian, mandiri, segmen yang tidak terintegrasi

Beberapa faktor dalam memilih strategi penyebaran

- Pembiayaan, otonomi, keamanan
- Pola situs data
- Pertumbuhan dan pengembangan
- Kemampuan teknologi
- Biaya pengelolaan teknologi
- Kebutuhan layanan yang dapat diandalkan

Strategi perancangan tersebar

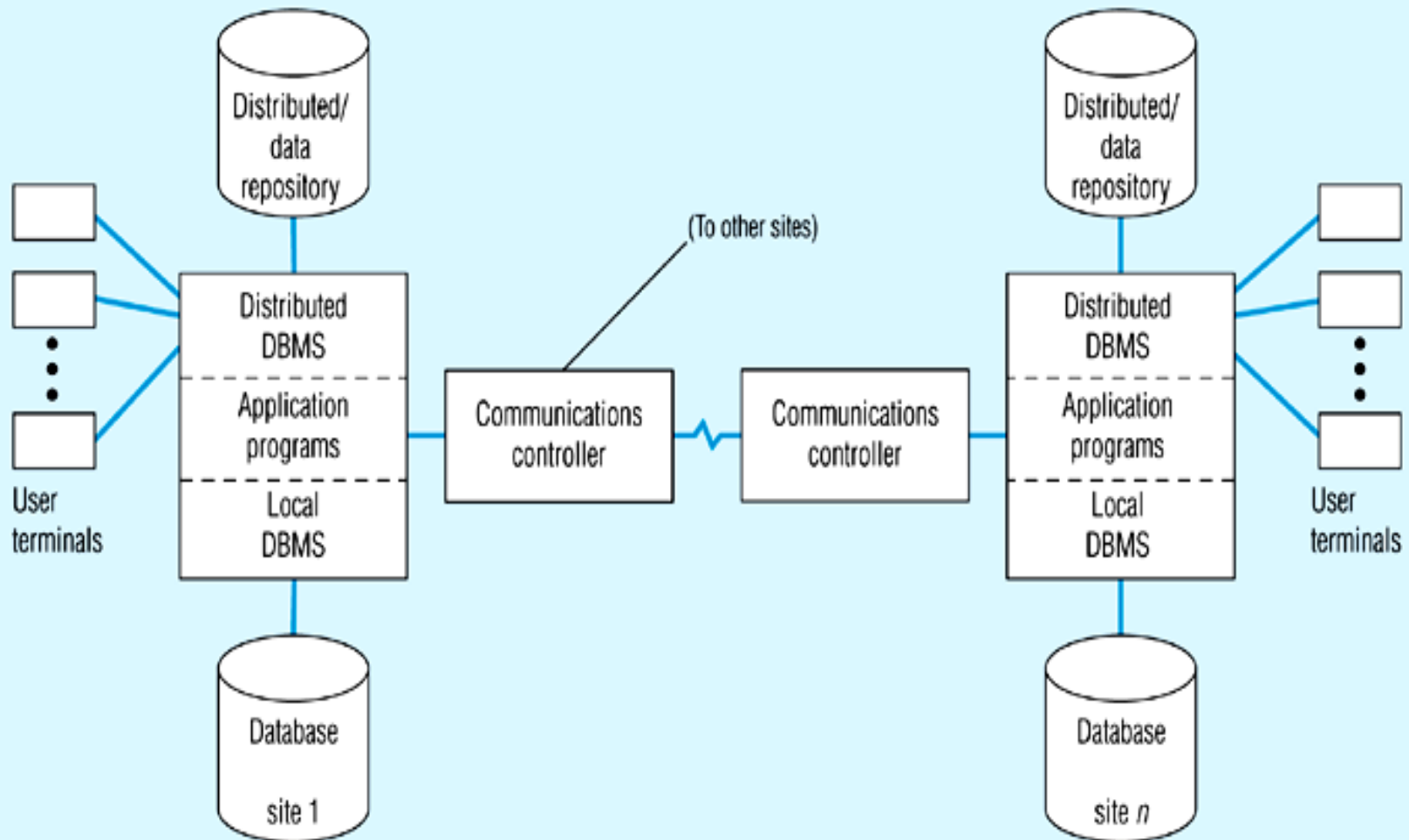
Table 13-1 Comparison of Distributed Database Design Strategies

<i>Strategy</i>	<i>Reliability</i>	<i>Expandability</i>	<i>Communications Overhead</i>	<i>Manageability</i>	<i>Data Consistency</i>
Centralized	POOR: Highly dependent on central server	POOR: Limitations are barriers to performance	VERY HIGH: High traffic to one site	VERY GOOD: One, monolithic site requires little coordination	EXCELLENT: All users always have same data
Replicated with snapshots	GOOD: Redundancy and tolerated delays	VERY GOOD: Cost of additional copies may be less than linear	LOW to MEDIUM: Not constant, but periodic snapshots can cause bursts of network traffic	VERY GOOD: Each copy is like every other one	MEDIUM: Fine as long as delays are tolerated by business needs
Synchronized replication	EXCELLENT: Redundancy and minimal delays	VERY GOOD: Cost of additional copies may be low and synchronization work only linear	MEDIUM: Messages are constant, but some delays are tolerated	MEDIUM: Collisions add some complexity to manageability	MEDIUM to VERY GOOD: Close to precise consistency
Integrated partitions	VERY GOOD: Effective use of partitioning and redundancy	VERY GOOD: New nodes get only data they need without changes in overall database design	LOW to MEDIUM: Most queries are local but queries which require data from multiple sites can cause a temporary load	DIFFICULT: Especially difficult for queries that need data from distributed tables, and updates must be tightly coordinated	VERY POOR: Considerable effort, and inconsistencies not tolerated
Decentralized with independent partitions	GOOD: Depends on only local database availability	GOOD: New sites independent of existing ones	LOW: Little if any need to pass data or queries across the network (if one exists)	VERY GOOD: Easy for each site, until there is a need to share data across sites	LOW: No guarantees of consistency, in fact pretty sure of inconsistency

Distributed DBMS

- *Database tersebar membutuhkan DBMS tersebar*
- Fungsi DBMS tersebar :
 - Meletakkan data dengan *kamus data terdistribusi*
 - Menentukan lokasi dari mana untuk mendapat data dan memproses komponen
 - DBMS menterjemahkan antar node dengan DBMS yang lain (menggunakan *middleware*)
 - Konsistensi data (melalui *multiphase commit protocols*)
 - Pengendalai kunci primer global
 - Scalability
 - Security, concurrency, optimasi query, perbaikan (recovery)

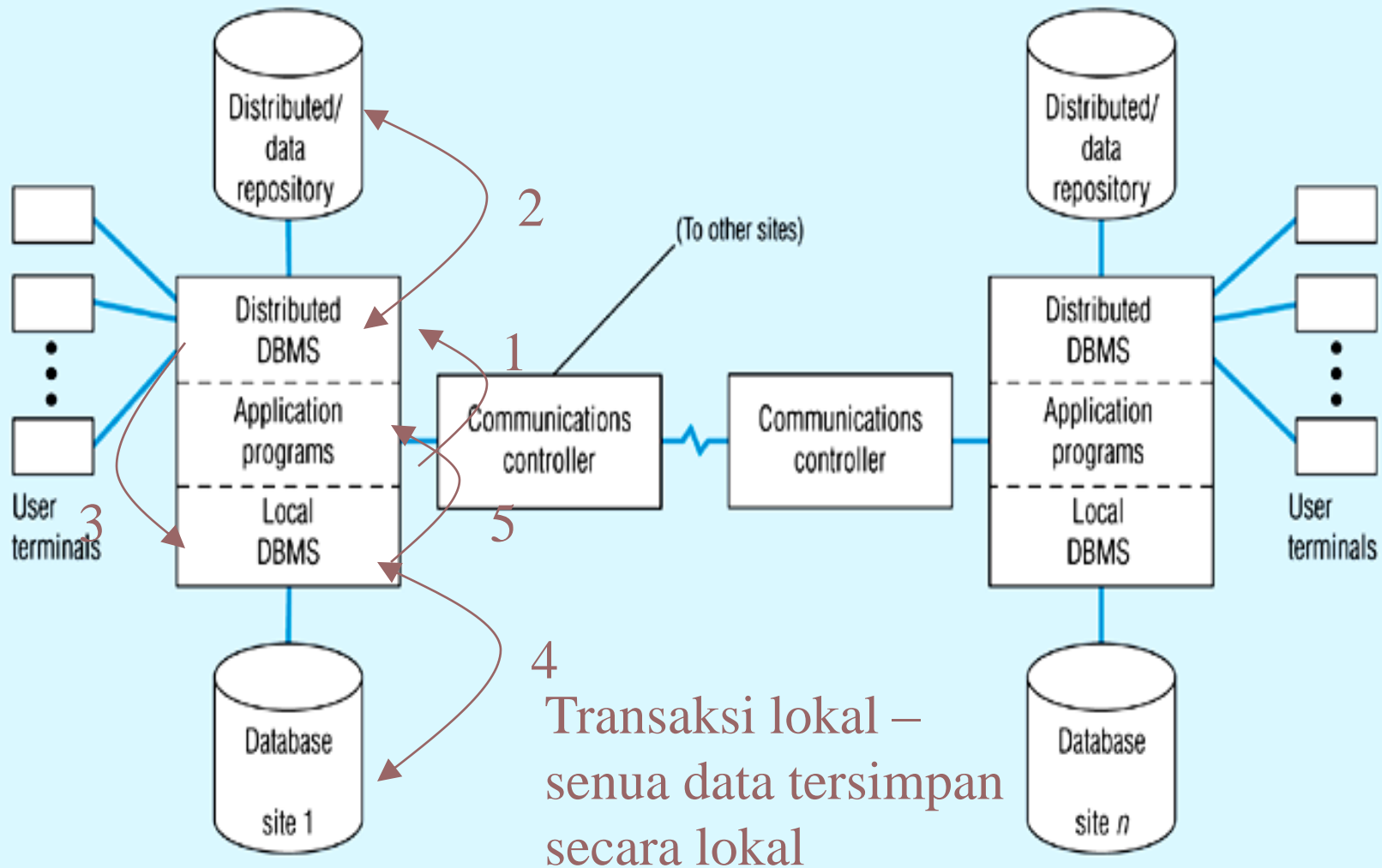
Arsitektur DBMS Tersebar



Langkah-langkah transaksi lokal

1. Aplikasi membuat permintaan kepada DBMS tersebar
2. DBMS tersebar melakukan pengecekan tempat penyimpanan data tersebar untuk menempatkan data. Pencarian ini bersifat **lokal**
3. DBMS terdistribusi mengirim permintaan ke DBMS lokal
4. DBMS Lokal mengolah permintaan
5. DBMS Lokal mengirim hasilnya ke aplikasi

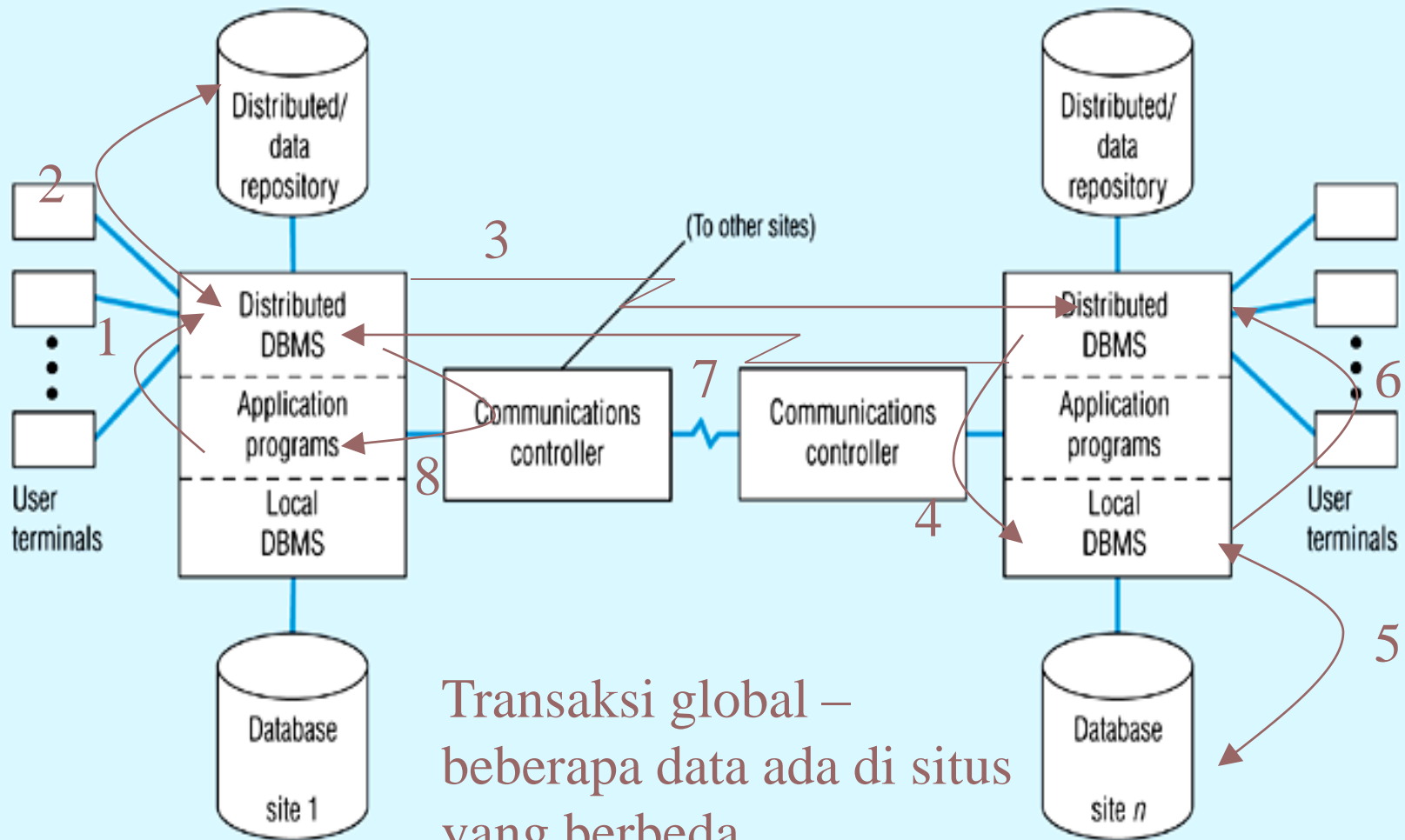
Arsitektur DBMS Tersebar yang menunjukkan Langkah-langkah transaksi Lokal



Langkah-langkah Transaksi Global

1. Aplikasi membuat permintaan kepada DBMS tersebar
2. DBMS tersebar melakukan pengecekan tempat penyimpanan data tersebar untuk menempatkan data. Pencarian bersifat **remote**
3. DBMS tersebar melanjutkan permintaan di situs lain
4. DBMS tersebar yang ada di situs lain menterjemahkan permintaan untuk DBMS lokal jika diperlukan, dan mengirimkan pesan ke DBMS lokal
5. DBMS lokal yang berada disuatu situs memproses permintaan
6. DBMS lokal mengirimkan hasilnya ke DBMS tersebar yang ada di situs yang lain
7. DBMS tersebar tersebut akan mengirimkan hasilnya kembali ke situs aslinya
8. DBMS tersebar di situs asal akan mengirimkan ke aplikasi

Arsitektur DBMS tersebar yang memperlihatkan langkah-langkah transaksi global



Tujuan transparansi DBMS terdistribusi

- **Transparansi Lokasi**
 - Pengguna/aplikasi tidak perlu tahu dimana data berada
- **Replication Transparency**
 - Pengguna/Aplikasi tidak ingin tahu adanya duplikasi
- **Transparansi Kesalahan**
 - Ada atau tidak setiap aksi suatu transaksi diterima
 - Setiap situs memiliki pengatur transaksi
 - Ada pencatatan transaksi sebelum dan sesudah
 - Pengendali konkurensi untuk menjamin integritas data
 - membutuhkan *protocol yang seragam*

Dua fase penerimaan

- ***Phase persiapan***

- Koordinator menerima permintaan persetujuan
- Koordinator meminta semua pengelola sumber daya untuk siap melakukan “sesuatu” terhadap transaksi. Setiap pengelola sumber daya mencatat semua updating dari suatu transaksi di log penyimpanan
- Koordinator menerima balasan dari semua pengelola sumber daya. Jika semuanya OK, catat penerimaan di masing-masing log; jika tidak maka tulis *rollback* di log.

Dua fase penerimaan

- ***Fase persetujuan***

- Koordinator memberi tahu setiap pengelola sumber daya tentang keputusan dan menyebarkan pesan untuk menerima atau mengulang (batal). Jika pesannya adakah menerima, kemudian setiap pengelola sumberdaya mengirim update dari log ke database-nya
- Suatu kesalahan dalam fase penerimaan mengakibatkan transaksi dibuang. Ini terjadi setelah melalui pengujian dan penanganan dengan batas waktu atau polling

Pengendalian konkurensi

- **Transparansi konkurensi**
 - Merancang tujuan dari database tersebar
- **Penandawaktu**
 - Mekanisme pengendalian konkurensi
 - Alternatif penguncian dalam database tersebar

Optimasi Query

- Dalam query yang membutuhkan penggabungan banyak situs dan, mungkin saja sebuah database tersebar yang terdiri dari replikasi data, DBMS tersebar harus membedakan dimana data yang akan diakses dan bagaimana melakukan proses dengan join.
- Tiga tahap proses:
 - 1 ***Dekomposisi query*** – penulisan kembali dan penyederhanaan
 - 2 ***Lokalisasi Data*** – query menggabungkan sebagian dari beberapa data kedalam satu tempat (situs)
 - 3 ***Optimalisasi Global*** -
 - Urutan sesuai dengan hasil query
 - Data bergerak antar situs
 - Dimana bagian dari query yang akan di eksekusi

Evolusi DBMS tersebar

- “Unit Kerja” – Semua langkah-langkah sebuah transaksi.
- Unit kerja jarak jauh
 - Pernyataan SQL originated at one location can be executed as a single unit of work on a single remote DBMS

Evolusi DBMS tersebar

- **Penyebaran Unit Kerja**
 - Berbagai pernyataan dalam suatu unit kerja mungkin menunjuk situs lain yang jauh
 - Semua database dalam pernyataan SQL tunggal harus ada dalam satu situs
- **Penyebaran Permintaan**
 - Sebuah pernyataan SQL tunggal mungkin menunjuk tabel-tabel yang berasal dari lebih satu situs
 - Mungkin tidak mendukung transparansi replikasi atau transparansi kerusakan