BASIS DATA TERDISTRIBUSI

Pendahuluan

Pada basis data terdistribusi (*distributed database*), data disimpan pada beberapa tempat (*site*), setiap tempat diatur dengan suatu DBMS (*Database Management System*).

Sistem Basis Data

Database Technology

Network Technology

Pengertian

- ***Distributed Computing Systems :***

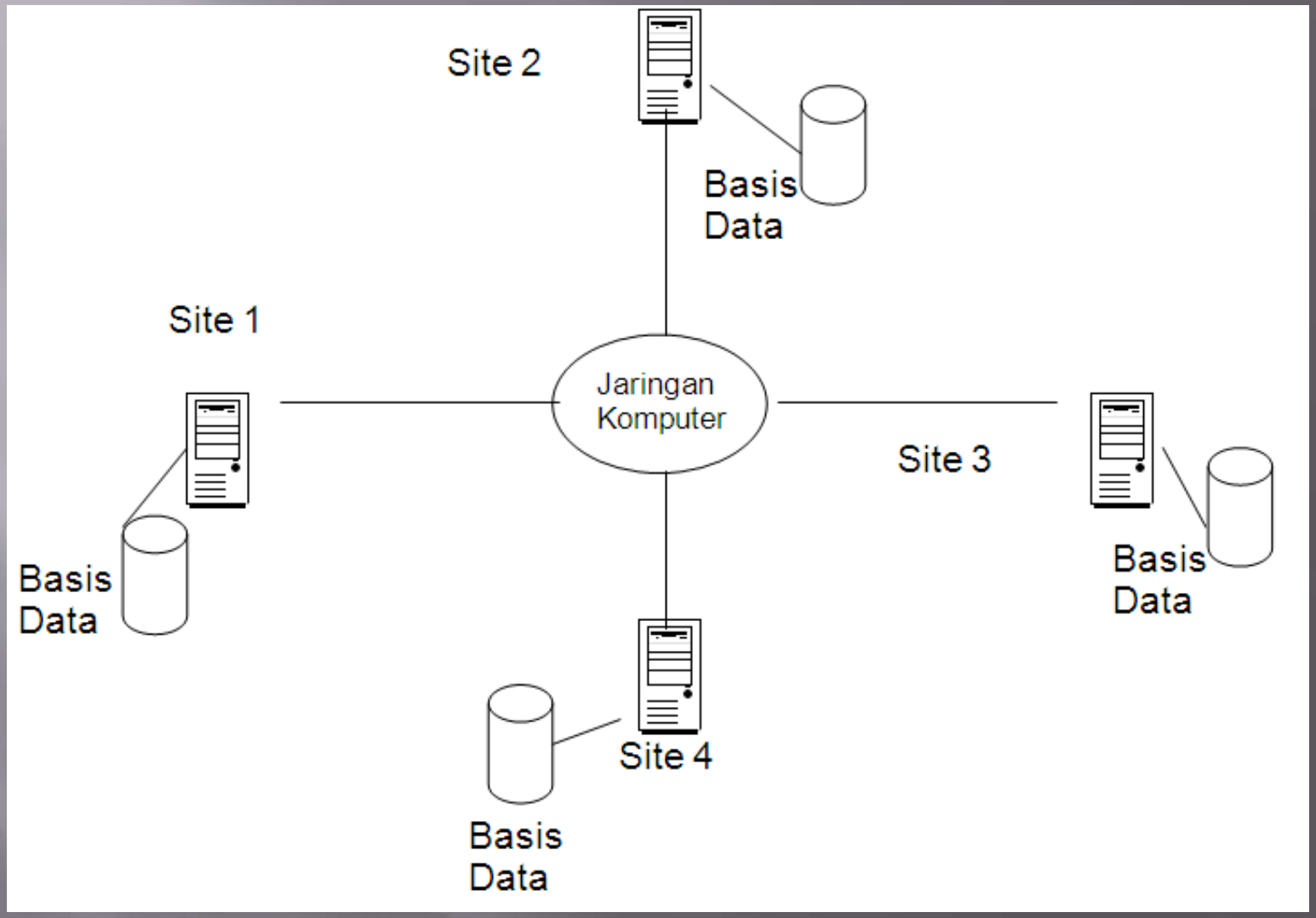
  Sejumlah unsur atau element pengolahan yang sifatnya otonom (autonomous), tidak harus homogen, yang saling terhubung melalui suatu jaringan komputer dan bekerjasama dalam melakukan tugasnya masing-masing.

- ***Distributed Database***

  Suatu kumpulan dari berbagai jenis basis data, yang saling terhubung dan terdistribusi melalui jaringan komputer.

- ***Distributed Database Management System***

  Software yang mengatur basis data terdistribusi, selama terjadinya distribusi transparan ke *user*.


Site 2
Basis
Data
Site 1
Jaringan
Komputer
Site 3
Basis
Data
Basis
Data
Site 4
Basis
Data

Contoh Basis Data Terdistribusi

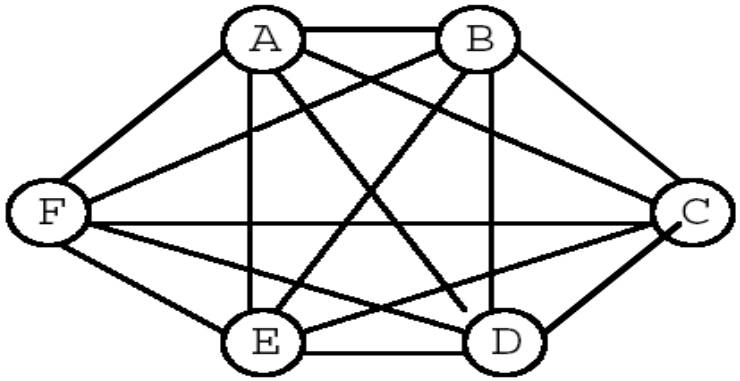
- Misalnya sebuah bank yang memiliki banyak cabang, bahkan di sebuah kota bisa terdiri dari beberapa cabang/kantor.
- Masing-masing lokasi memiliki jaringan lokal sendiri, dan semua jaringan lokal itu dihubungkan satu sama lain membentuk sebuah jaringan nasional.

Topologi Basis Data Terdistribusi

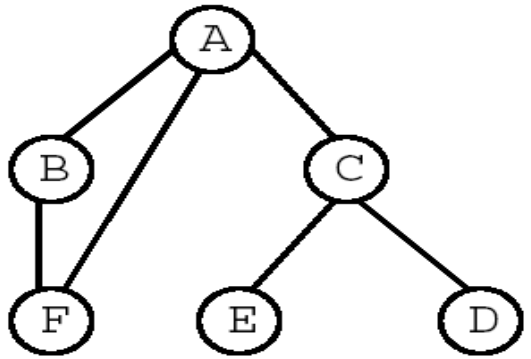
Sebuah sistem basis data terdistribusi hanya mungkin dibangun dalam sebuah jaringan komputer.

A. *Fully Connected Network*
B. *Partially Connected Network*
C. *Tree Structured Network*
D. *Ring Network*
E. *Star Network*

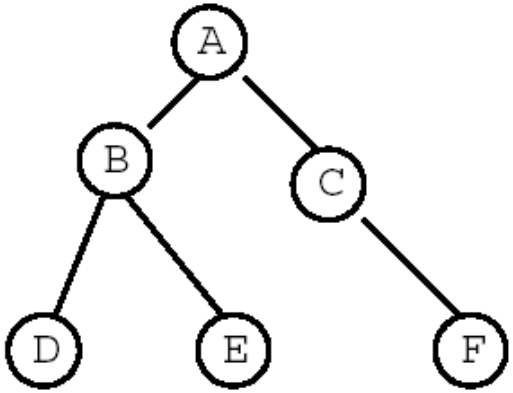
Topologi Basis Data Terdistribusi



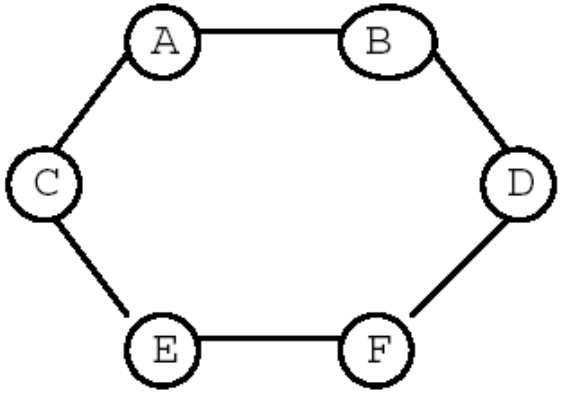
Fully connected network



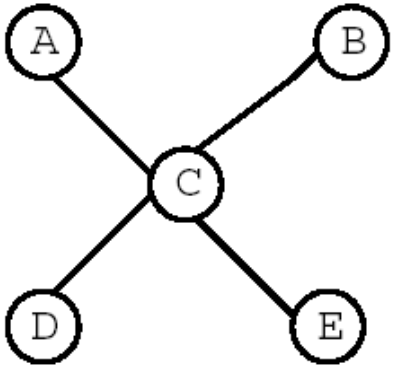
Partially connected network



Tree structured network



Ring network



Star network

Transaksi Lokal dan Global

Ada 2 jenis transaksi yang mungkin terjadi:

1. **Transaksi Lokal**

   Sebuah transaksi yang mengakses basis data di *server* yang sama dengan *server* dari mana transaksi tersebut dijalankan.

2. **Transaksi Global**

   Transaksi yang mengakses data pada *server* yang berbeda dengan *server* dimana transaksi tersebut dijalankan atau transaksi dari *server* yang membutuhkan akses data ke sejumlah *server* lain.

Contoh Transaksi

- Transaksi untuk menambahkan 50 juta pada nomor rekening 177 yang berada di cabang A.
- Jika transaksi telah ditentukan pada cabang A, maka transaksi ini dianggap transaksi lokal.
- Jika sebuah transaksi untuk mentransfer 50 juta dari rekening 177 ke rekening 305 yang berlokasi di cabang B, maka transaksi ini dikatakan transaksi global karena rekening di dua *site* yang berbeda telah diakses sebagai hasil dari eksekusinya.

Keuntungan Basis Data Terdistribusi

1. **Pengawasan distribusi dan pengambilan data**

   Jika sejumlah *site* yang berbeda dihubungkan satu sama lain, lalu seorang pemakai yang berada pada satu *site* dapat mengakses data yang tersedia pada *site* lain.

   Sebagai contoh: sistem distribusi pada sebuah bank memungkinkan seorang pemakai pada salah satu cabang dapat mengakses data cabang lain.

2. **Reliability dan availability**

   Sistem distribusi dapat terus menerus berfungsi dalam menghadapi kegagalan dari *site* individu atau mata rantai komunikasi antar *site*.

   Misal: jika *site-site* gagal dalam sebuah sistem distribusi, *site-site* lainnya dapat melanjutkan operasi jika data telah direplikasi pada beberapa *site*.

Keuntungan Basis Data Terdistribusi

3. **Kecepatan pemrosesan query**

   Jika sebuah *query* melibatkan data pada beberapa *site*, memungkinkan membagi *query* ke dalam *sub query* yang dapat dieksekusi dalam bentuk paralel oleh beberapa *site*. Perhitungan secara paralel mempercepat pemrosesan dari seorang pemakai *query*.

4. **Otonomi lokal**

   Pendistribusian sistem mengizinkan sekelompok individu dalam sebuah perusahaan untuk melatih pengawasan lokal melalui data mereka sendiri. Dengan kemampuan ini dapat mengurangi ketergantungan pada pusat pemrosesan.

5. **Efisien dan fleksibel**

   Data dalam sistem distribusi dapat disimpan dekat dengan titik di mana data tersebut dipergunakan. Data dapat secara dinamik bergerak atau disalin, atau salinannya dapat dihapus.

Kerugian Basis Data Terdistribusi

1. **Harga *software* yang mahal**
   Hal ini disebabkan sangat sulit untuk membuat sistem basis data terdistribusi.

2. **Kemungkinan kesalahan lebih besar**
   Site-site yang termasuk dalam sistem terdistribusi beroperasi secara paralel sehingga menjadi lebih sulit untuk menjamin kebenaran dari algoritma. Adanya kesalahan mungkin tak dapat diketahui.

3. **Biaya pemrosesan tinggi**
   Perubahan pesan-pesan dan penambahan perhitungan dibutuhkan untuk mencapai koordinasi antar *site*.

Desain Basis Data pada Sistem Terdistribusi

Pada basis data terdistribusi, relasi dapat disimpan pada beberapa tempat.

A. **FRAGMENTASI**

   Fragmentasi terdiri dari relasi yang dibagi ke relasi atau fragmen yang lebih kecil dan mengirim fragmen pada beberapa tempat.

B. **REPLIKASI**

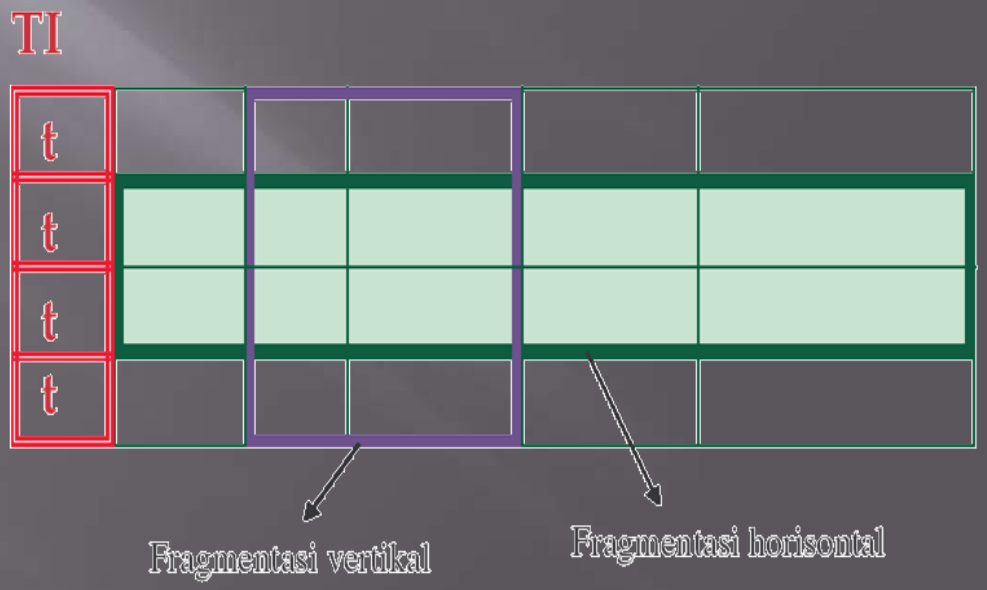
   Sistem memelihara sejumlah salinan/duplikat tabel-tabel data. Setiap salinan tersimpan dalam simpul yang berbeda, yang menghasilkan replikasi data.

FRAGMENTASI

- Fragmentasi data memisahkan relasi ke dalam beberapa fragment. Tiap-tiap fragment disimpan pada *site* yang berbeda.

3 Jenis Fragmentasi:

1. Fragmentasi Horizontal
2. Fragmentasi Vertical
3. Fragmentasi Campuran (*Hybrid*)



FRAGMENTASI HORIZONTAL

- Fragmentasi horizontal berisikan *tuple-tuple* yang dipartisi dari sebuah relasi global ke dalam sejumlah subset r1, r2, ... , rn.
- Tiap-tiap subset berisikan sejumlah *tuple* dari r. Tiap-tiap *tuple* dari r harus memiliki satu fragment, sehingga relasi yang asli dapat disusun kembali.
- Sebuah fragment dalam fragmentasi horizontal dapat didefinisikan sebagai sebuah seleksi pada relasi global r.
- Oleh karena itu sebuah predikat Pi digunakan untuk menyusun fragment ri seperti berikut :

$$ri = \sigma i(r)$$

- Penyusunan kembali dari relasi r dapat diperoleh dengan mengambil gabungan dari seluruh fragment.

FRAGMENTASI VERTICAL

- Dalam fragmentasi vertikal, tiap-tiap fragment ri didefinisikan sebagai :

$$ri = \Pi i(r)$$

- Relasi global dapat disusun kembali dari fragment-fragment dengan mengambil natural join:

$$r = r1\ r2\ r\ .....\ Rn$$

- Fragmentasi vertikal disempurnakan dengan menambahkan sebuah atribut yang disebut *tuple identifier* (tuple-id) ke dalam skema r. Sebuah *tuple-id* adalah sebuah alamat logik dari sebuah *tuple*.
- Tiap-tiap *tuple* dalam r harus memiliki sebuah alamat yang unik, atribut *tuple-id* sebagai kunci untuk penambahan skema.

FRAGMENTASI CAMPURAN

- Relasi r (global) dibagi-bagi ke dalam sejumlah relasi fragment r1, r2, r3, ..., rn.
- Tiap-tiap fragment diperoleh sebagai hasil baik dari skema fragmentasi horizontal ataupun skema fragmentasi vertikal pada relasi r, atau dari sebuah fragment r yang diperoleh sebelumnya.

Cara membangun fragmentasi campuran :

a. Menggunakan fragmentasi horizontal pada fragmentasi vertikal.
b. Menggunakan fragmentasi vertikal pada fragmentasi horizontal.

Contoh Fragmentasi

Relasi :

- Deposit (branch_name, account_number, customer_name, balance)

Branch-name	account-number	Customer-name	balance
Hillside	305	Lowman	500
Hillside	226	Camp	336
Valleyview	177	Camp	205
Valleyview	402	Khan	10000
Hillside	115	Khan	62
Valleyview	408	Khan	1123
Valleyview	639	Green	750

Horizontal

- Jika bank hanya memiliki dua cabang, Hillside dan Valleyview maka ada dua fragment yang berbeda.

- Kemudian fragmentasi horizontal dapat diuraikan sebagai berikut :

Deposit1 = σ branch-name = "Hillside“ (Deposit)

Deposit2 = σ branch-name = "Valleyview" (Deposit)

Fragment deposit1 disimpan pada site Hillside dan fragment deposit2 disimpan pada site Valleyview.

Horizontal

Dua Fragmen yang diperoleh :

branch-name	account-number	customer-name	balance
Hillside	305	Lowman	500
Hillside	226	Camp	336
Hillside	115	Khan	62

(a) $deposit_1$

branch-name	account-number	customer-name	balance
Valleyview	177	Camp	205
Valleyview	402	Khan	10000
Valleyview	408	Khan	1123
Valleyview	639	Green	750

(b) $deposit_2$

Horizontal

Kondisi penyusunan kembali sangat mudah untuk diperiksa karena selalu mungkin untuk disusun kembali relasi global deposit melalui operasi union sebagai berikut:

Deposit = deposit1 U deposit2

Vertical

- Pada fragmentasi vertikal, relasi deposit memerlukan penambahan *tuple-id*.
- Berikut ini adalah relasi deposit dengan penambahan *tuple-id*:

branch-name	account-number	customer-name	balance	tuple-id
Hillside	305	Lowman	500	1
Hillside	226	Camp	336	2
Valleyview	177	Camp	205	3
Valleyview	402	Khan	10000	4
Hillside	115	Khan	62	5
Valleyview	408	Khan	1123	6
Valleyview	639	Green	750	7

Vertical

- Sebuah fragmentasi vertikal dari relasi ini dapat diuraikan sebagai berikut :

Deposit3 = π branch-name,customer-name,tuple-id (deposit)
Deposit4 = π account-number,balance,tuple-id (deposit)

branch-name	customer-name	tuple-id
Hillside	Lowman	1
Hillside	Camp	2
Valleyview	Camp	3
Valleyview	Khan	4
Hillside	Khan	5
Valleyview	Khan	6
Valleyview	Green	7

(a) relasi $deposit_3$

account-number	Balance	tuple-id
305	500	1
226	336	2
177	205	3
402	10000	4
115	62	5
408	1123	6
639	750	7

(b) relasi $deposit_4$

Vertical

- Untuk menyusun kembali relasi deposit yang asli dari fragment-fragment, dapat menggunakan:

  Π Deposit-scheme(deposit3 deposit4)

- Atribut join dari ekspresi di atas adalah ***tuple-id***. Karena *tuple-id* menggambarkan sebuah alamat, hal ini memungkinkan untuk memasangkan sebuah *tuple* dari deposit3 yang berhubungan dengan *tuple* dari deposit4 dengan menggunakan alamat yang diberikan oleh harga *tuple-id*.

Campuran

- Misalkan relasi r adalah relasi deposit dari contoh sebelumnya. Relasi ini dibagi ke dalam fragment deposit3 dan deposit4 (Vertical).

- Selanjutnya kita dapat membagi fragment deposit3 menjadi fragment deposit3a dan fragment deposit3b dengan menggunakan skema fragmentasi horizontal ke dalam dua fragment berikut :

Deposit3a= σ branch-name = "Hillside" (Deposit3)

Deposit3b= σ branch-name = "Valleyview" (Deposit3)

Campuran

branch-name	customer-name	tuple-id
Hillside	Lowman	1
Hillside	Camp	2
Hillside	Khan	5

(a)Relasi $deposit_{3a}$

branch-name	customer-name	tuple-id
Valleyview	Camp	3
Valleyview	Khan	4
Valleyview	Khan	6
Valleyview	Green	7

(b) relasi $deposit_{3b}$