

## PERKULIAHAN KE 13

### BASIS DATA TERDISTRIBUSI

#### 10.1 Pengertian Basis Data Terdistribusi

Tujuan utama di balik perkembangan sistem basis data adalah suatu keinginan untuk mengintegrasikan berbagai data-data operasional dari suatu organisasi dan menyediakan pengaksesan data yang terkontrol. Walaupun integrasi dan pengontrolan pengaksesan data akan mengimplikasikan suatu sentralisasi data dan sistem. Pada kenyataannya perkembangan jaringan computer mengarah kepada model kerja yang desentralisasi. Desentralisasi tersebut akan memperkenalkan struktur organisasi banyak perusahaan, yang secara logis terdistribusi ke dalam divisi-divisi, departemen-departemen, proyek-proyek, dan lain-lain, dan secara fisik terdistribusi ke dalam offices (kantor-kantor), plants (bangunan-bangunan untuk pekerjaan produksi suatu perusahaan), factories (pabrik-pabrik), dan lain-lain, dimana masing-masing unit akan merancang dan membiayai pengolahan data mereka masing-masing.

Perkembangan sistem basis data terdistribusi akan merefleksikan dan mencerminkan struktur organisasional tersebut, membuat data diseluruh unit dapat diakses dengan baik dan menyimpan data-data penting dan sering digunakan ke tempat-tempat yang paling sering digunakan dan mudah ditemukan, serta meningkatkan kemampuan data untuk digunakan secara bersama-sama berikut efisiensi pengaksesannya.

Menurut (Ladjamudin, 2004) menyampaikan bahwa Basis Data Terdistribusi yaitu kumpulan data logic yang saling berhubungan, secara fisik terdistribusi dalam

jaringan komputer, yang tidak tergantung dari program aplikasi, dan dapat digunakan oleh banyak aplikasi sekarang maupun pada masa yang akan datang. DBMS terdistribusi adalah sistem software yang memungkinkan ditatanya suatu basis data terdistribusi bagi setiap pemakai (user). Menurut (Ladjamudin, 2004), DBMS terdistribusi memiliki beberapa karakteristik antara lain:

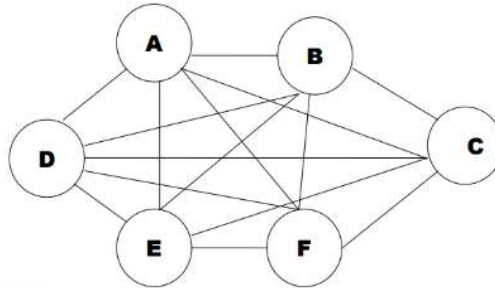
1. Kumpulan data-data logic (yang dapat digunakan secara bersama) terdistribusi pada beberapa unit computer yang berbeda.
2. Komputer tersebut terkoneksi ke dalam suatu jaringan komunikasi
3. Data pada masing-masing unit computer (work-station) terkontrol oleh suatu DBMS.
4. DBMS pada masing-masing bagian dapat menangani aplikasi-aplikasi local, secara otomatis.
5. Masing-masing DBMS berpartisipasi paling tidak pada satu aplikasi global.

## **10.2 Topologi Distribusi Data**

Menurut (Fathansyah, 2012), sebuah sistem basis data terdistribusi hanya mungkin dibangun dalam sebuah sistem jaringan komputer. Dalam sebuah sistem jaringan computer kita mengenal adanya Topologi., yang akan menentukan bagaimana konfigurasi/ keterhubungan antara satu simpul jaringan (node/site) dengan simpul-simpul lainnya. Setiap simpul, dalam kaitannya dengan sistem basis data terdistribusi mewakili sebuah server, yang memiliki disk dengan sistem data sendiri (lokal). Setiap server ini juga membuat sebuah LAN (Local area Network) sendiri untuk mengamodasi sejumlah workstation dan sekaligus user lokal.

Bentuk-Bentuk Topologi Terdistribusi :

1. *Fully Connected Network*



Gambar 10.1  
*Fully Connected Network*

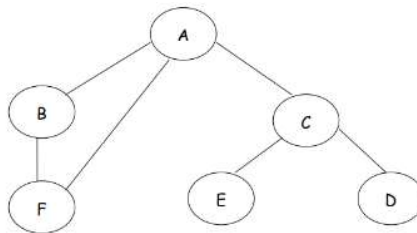
Keuntungan:

Kalau salah satu node rusak, yang lainnya masih dapat berjalan

Kerugian :

Control manajemen tidak terjamin

2. *Partially Connected Network*



Gambar 10.2  
*Partially Connected Network*

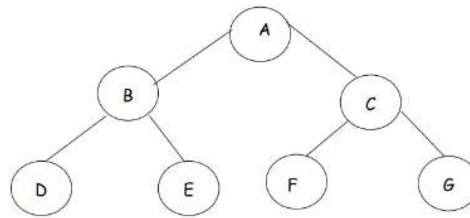
Keuntungan:

- a. Reliability rendah
- b. Biaya dapat ditekan

Kerugian :

Control manajemen tidak terjamin

3. *Tree Structured network*



Gambar 10.3  
*Tree Structured Network*

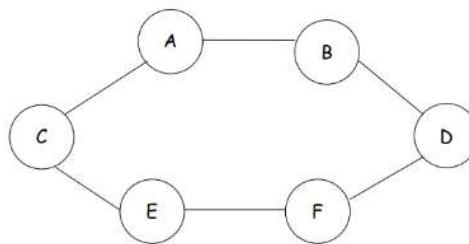
Keuntungan:

- a. Bersifat sentral
- b. Control manajemen terjamin

Kerugian :

Kalau node pusat rusak maka semua akan rusak

4. *Ring Network*



Gambar 10.4  
*Ring Network*

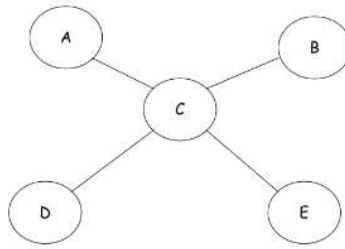
Keuntungan :

Apabila ada satu node yang rusak maka yang lain masih berjalan

Kerugian :

Control manajemen kurang terjamin karena bersifat desentralisasi.

5. *Star Network*



Gambar 10.5  
*Star Network*

Keuntungan:

Control manajemen lebih terjamin, karena bersifat sentral dan reliability rendah

Kerugian:

Kalau ada node pusat rusak maka yang lainnya rusak juga.

### 10.3 Keuntungan dan Kerugian Basis Data Terdistribusi

1. Keuntungan Basis Data Terdistribusi

- a. Secara alami mengikuti struktur organisasi
- b. Adanya otonomi lokal
- c. Sifatnya dapat dipakai secara bersama
- d. Peningkatan ketersediaan
- e. Peningkatan kehandalan
- f. Peningkatan kinerja
- g. Ekonomis
- h. Pertumbuhan yang modular

2. Kerugian Basis Data Terdistribusi
  - a. Harga software mahal (Biaya)
  - b. Kompleksitas
  - c. Kelemahan dalam keamanan
  - d. Sulitnya menjaga keutuhan data
  - e. Kurangnya standar
  - f. Kurangnya pengalaman
  - g. Perancangan basisdata lebih kompleks

#### **10.4 Fragmentasi Data**

##### **1. Pengertian Fragmentasi**

Fragmentasi merupakan sebuah proses pembagian atau pemetaan database dimana database dipecah-pecah berdasarkan kolom dan baris yang kemudian disimpan didalam site atau unit komputer yang berbeda dalam suatu jaringan data, sehingga memungkinkan untuk pengambilan keputusan terhadap data yang telah terbagi. Fragmentasi data merupakan langkah yang diambil untuk menyebarkan data dalam basis data terdistribusi.

Menurut (Hariyanto, 2004), alasan-alasan diperlukannya fragmentasi, yaitu :

##### **a. Penggunaan**

Umumnya, aplikasi-aplikasi beroperasi dengan terhadap suatu view tertentu bukan seluruh relasi. Dengan demikian untuk melakukan distribusi data maka beralasan untuk bekerja dengan suatu subset relasi (fragmen).

b. Efisiensi

Data yang disimpan dekat dengan aplikasi yang sering menggunakannya.

Data yang tidak diperlukan oleh aplikasi local tidak disimpan di situs itu.

c. Parallelsisme

Dengan fragmen-fragmen sebagai unit distribusi, transaksi dapat dibagi menjadi beberapa subquery yang beroperasi pada fragmen-fragmen itu.

Fragmentasi harus meningkatkan derajat konkurensi atau paralelisme sistem.

d. Keamanan

Data yang tidak diperlukan oleh aplikasi local tidak disimpan di situs itu.

Dengan cara ini, data tidak tersedia untuk pemakai-pemakai yang tidak diotorisasi.

2. Aturan Fragmentasi

Beberapa Peraturan Yang Harus Didefinisikan Ketika Mendefinisikan Fragment, adalah :

a. Kondisi lengkap (*Completeness* )

Sebuah unit data yang masih dalam bagian dari relasi utama, maka data harus berada dalam satu fragmen. Ketika ada relasi, pembagian datanya harus menjadi satu kesatuan dengan relasinya.

b. Rekontruksi (*Reconstruction* )

Sebuah relasi asli dapat dibuat kembali atau digabungkan kembali dari sebuah fragmen. Ketika telah dipecah-pecah, data masih memungkinkan untuk digabungkan kembali dengan tidak mengubah struktur data.

c. *Disjointness*

Data didalam fragmen tidak boleh diikutkan dalam fragmen lain agar tidak terjadi redundancy data, kecuali untuk atribut primary key dalam fragmentasi vertical

Menurut (Hariyanto, 2004), kerugian fragmentasi yaitu :

a. Kinerja

Kinerja aplikasi yang memerlukan data dari fragmen-fragmen yang berlokasi terpisah dapat lebih lambat.

b. Integritas

Kendali integritas dapat lebih sulit jika data dan kebergantungan fungsional difragmentasi dan berlokasi di situs-situs yang berbeda.

3. Jenis-Jenis Fragmentasi

a. Fragmentasi horizontal

Terdiri dari tuple dari fragment global yang kemudian dipecah-pecah atau disekat menjadi beberapa sub-sets. Fragmentasi horizontal berisikan tuple2 yang dipartisikan dari sebuah relasi global ke dalam sejumlah subset  $r_1, r_2, \dots, r_n$ , tiap2 subset berisi tuple dari  $r$ , setiap tuple dari  $r$  harus memiliki satu fragment sehingga relasi yang asli dapat disusun kembali.



b. Fragmentasi vertikal

Membagi atribut-atribut dari fragment global yang tersedia menjadi beberapa grup. Penambahan tuple-id didalam fragmentasi vertikal. 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. Setiap tuple didalam r harus memiliki sebuah alamat yang unik, yaitu attribute tuple-id sebagai kunci penambahan skema.

c. Fragmentasi campuran

Relasi r (global) dibagi2 kedalam sejumlah relasi fragment  $r_1, r_2, \dots, r_n$ . Tiap2 fragmentasi diperoleh sebagai hasil baik dari skema fragmentasi horizontal ataupun fragmentasi vertikal di relasi r atau dari sebuah fragmentasi r yang diperoleh sebelumnya.

Cara yang sederhana untuk membangun fragmentasi campuran sbb :

- 1) Menggunakan fragmentasi horizontal pada fragmentasi vertikal
- 2) Menggunakan fragmentasi vertical pada fragmentasi horizontal

Contoh Penggunaan Fragmentasi :

Kasus Jenis-Jenis Fragmentasi Ujian (NIM, Nama\_Mhs, Kode\_MK, Mt\_Kuliah, Nil\_Akhir, Grade )

| NIM | Nama_Mhs  | Kode_MK | Mt_Kuliah         | Nil_Akhir | Grade |
|-----|-----------|---------|-------------------|-----------|-------|
| 123 | Fathi     | 101     | Sistem Basis Data | 78        | B     |
| 124 | Farah     | 102     | Peranc. Sistem    | 60        | C     |
| 125 | Sarah     | 101     | Sistem Basis Data | 40        | D     |
| 126 | Salsabila | 101     | Sistem Basis Data | 90        | A     |
| 127 | Azizah    | 103     | Visual Basic      | 70        | B     |
| 128 | Farhan    | 103     | Visual Basic      | 40        | D     |
| 129 | Faiz      | 102     | Peranc. Sistem    | 80        | A     |

Fragmentasi Horizontal terbagi menjadi 3 fragment yang berbeda berdasarkan Mt\_Kuliah

a) Relasi Mt\_Kuliah="Sistem Basis Data"

$\sigma$  Mt\_Kuliah="Sistem Basis Data" (Ujian)

| NIM | Nama_Mhs  | Kode_MK | Mt_Kuliah         | Nil_Akhir | Grade |
|-----|-----------|---------|-------------------|-----------|-------|
| 123 | Fathi     | 101     | Sistem Basis Data | 78        | B     |
| 125 | Sarah     | 101     | Sistem Basis Data | 40        | D     |
| 126 | Salsabila | 101     | Sistem Basis Data | 90        | A     |

b) Relasi Mt\_Kuliah="Peranc. Sistem"

$\sigma$  Mt\_Kuliah="Peranc. Sistem" (Ujian)

| NIM | Nama_Mhs | Kode_MK | Mt_Kuliah      | Nil_Akhir | Grade |
|-----|----------|---------|----------------|-----------|-------|
| 124 | Farah    | 102     | Peranc. Sistem | 60        | C     |
| 129 | Faiz     | 102     | Peranc. Sistem | 80        | A     |

c) Relasi Mt\_Kuliah="Visual Basic"

$\sigma$  Mt\_Kuliah="Visual Basic" (Ujian)

| NIM | Nama_Mhs | Kode_MK | Mt_Kuliah    | Nil_Akhir | Grade |
|-----|----------|---------|--------------|-----------|-------|
| 127 | Azizah   | 103     | Visual Basic | 70        | B     |
| 128 | Farhan   | 103     | Visual Basic | 40        | D     |

### Contoh Fragmentasi Vertical

Fragment di atas memenuhi kondisi jika Nama\_Mhs dan Mt\_Kuliah adalah hal-hal yang memenuhi syarat Fragmentasi vertical: berdasarkan dekomposisi-nya dengan menambahkan tuple\_id

| NIM | Nama_Mhs  | Kode_MK | Mt_Kuliah      | Nil_Akhir | Grade | Tuple_ID |
|-----|-----------|---------|----------------|-----------|-------|----------|
| 123 | Fathi     | 101     | Sistem Basis   | 78        | B     | 1        |
| 124 | Farah     | 102     | Data           | 60        | C     | 2        |
| 125 | Sarah     | 101     | Peranc. Sistem | 40        | D     | 3        |
| 126 | Salsabila | 101     | Sistem Basis   | 90        | A     | 4        |
| 127 | Azizah    | 103     | Data           | 70        | B     | 5        |
| 128 | Farhan    | 103     | Sistem Basis   | 40        | D     | 6        |
| 129 | Faiz      | 102     | Data           | 80        | A     | 7        |
|     |           |         | Visual Basic   |           |       |          |
|     |           |         | Visual Basic   |           |       |          |
|     |           |         | Peranc. Sistem |           |       |          |

Relasi 1 = NIM, Nama\_Mhs, Mt,Kuliah, Nil\_Akhir, Grade, Tuple\_ID

$\pi$  NIM,Nama\_Mhs,Mt,Kuliah,Nil\_Akhir,Grade,Tuple\_ID (UJian )

| NIM | Nama_Mhs  | Mt_Kuliah         | Nil_Akhir | Grade | Tuple_ID |
|-----|-----------|-------------------|-----------|-------|----------|
| 123 | Fathi     | Sistem Basis Data | 78        | B     | 1        |
| 124 | Farah     | Peranc. Sistem    | 60        | C     | 2        |
| 125 | Sarah     | Sistem Basis Data | 40        | D     | 3        |
| 126 | Salsabila | Sistem Basis Data | 90        | A     | 4        |
| 127 | Azizah    | Visual Basic      | 70        | B     | 5        |
| 128 | Farhan    | Visual Basic      | 40        | D     | 6        |
| 129 | Faiz      | Peranc. Sistem    | 80        | A     | 7        |

Relasi 2 = NIM,Kode\_MK,Nil\_Akhir,Grade,Tuple\_ID

$\pi$  NIM,Kode\_MK,Nil\_Akhir,Grade,Tuple\_ID (Ujian)

| NIM | Kode_MK | Nil_Akhir | Grade | Tuple_ID |
|-----|---------|-----------|-------|----------|
| 123 | 101     | 78        | B     | 1        |
| 124 | 102     | 60        | C     | 2        |
| 125 | 101     | 40        | D     | 3        |
| 126 | 101     | 90        | A     | 4        |
| 127 | 103     | 70        | B     | 5        |
| 128 | 103     | 40        | D     | 6        |
| 129 | 102     | 80        | A     | 7        |

### Contoh Fragmentasi Campuran

Terdapat relasi berdasarkan Mata Kuliah yang sama

Relasi 1a.

$\pi$  NIM, Nama\_Mhs, Mt\_Kuliah, Nil\_Akhir, Grade, Tuple\_ID (  $\sigma$  Mt\_Kuliah = “Sistem Basis Data” (Ujian))

| NIM | Nama_Mhs  | Mt_Kuliah         | Nil_Akhir | Grade | Tuple_ID |
|-----|-----------|-------------------|-----------|-------|----------|
| 123 | Fathi     | Sistem Basis Data | 78        | B     | 1        |
| 125 | Sarah     | Sistem Basis Data | 40        | D     | 3        |
| 126 | Salsabila | Sistem Basis Data | 90        | A     | 4        |

Relasi 1b.

$\pi$  NIM, Nama\_Mhs, Mt\_Kuliah, Nil\_Akhir, Grade, Tuple\_ID(  $\sigma$  Mt\_Kuliah= “Peranc. Sistem” (Ujian))

| NIM | Nama_Mhs | Mt_Kuliah      | Nil_Akhir | Grade | Tuple_ID |
|-----|----------|----------------|-----------|-------|----------|
| 124 | Farah    | Peranc. Sistem | 60        | C     | 2        |
| 129 | Faiz     | Peranc. Sistem | 80        | A     | 7        |

Relasi 1c

$\pi$  NIM, Nama\_Mhs, Mt\_Kuliah, Nil\_Akhir, Grade, Tuple\_ID(  $\sigma$  Mt\_Kuliah= “Visual Basic” (Ujian))

| NIM | Nama_Mhs | Mt_Kuliah    | Nil_Akhir | Grade | Tuple_ID |
|-----|----------|--------------|-----------|-------|----------|
| 127 | Azizah   | Visual Basic | 70        | B     | 5        |
| 128 | Farhan   | Visual Basic | 40        | D     | 6        |