**PERTEMUAN KEENAM, ALTERNATIF DESAIN MANAJEMEN SISTEM BASIS DATA TERDISTRIBUSI**

**Non-replicated & Non-fragmented**

1. Tabel berbeda di tempatkan di site yang berbeda.
2. Cocok digunakan untuk sistem basis data yang presentasi kebutuhan menggabungkan informasi dalam sebuah tabel yang berbeda site sangat rendah.
3. Mengurangi biaya komunikasi dalam proses pengolahan data.

In this design alternative, different tables are placed at different sites. Data is placed so that it is at a close proximity to the site where it is used most. It is most suitable for database systems where the percentage of queries needed to join information in tables placed at different sites is low. If an appropriate distribution strategy is adopted, then this design alternative helps to reduce the communication cost during data processing.

**Fully Replicated**

1. Di setiap site, terdapat salinan basis data dari site lain.
2. Proses query sangat cepat, membutuhkan sedikit biaya komunikasi.
3. Kekurangannya adalah data ganda dalam basis data membutuhkan biaya yang besar ketika perubahan data.

In this design alternative, at each site, one copy of all the database tables is stored. Since, each site has its own copy of the entire database, queries are very fast requiring negligible communication cost. On the contrary, the massive redundancy in data requires huge cost during update operations. Hence, this is suitable for systems where a large number of queries is required to be handled whereas the number of database updates is low.

**Partially Replicated**

1. Salinan tabel atau bagian dari tabel ditempatkan di site yang berbeda.

Copies of tables or portions of tables are stored at different sites. The distribution of the tables is done in accordance to the frequency of access. This takes into consideration the fact that the frequency of accessing the tables vary considerably from site to site. The number of copies of the tables (or portions) depends on how frequently the access queries execute and the site which generate the access queries.

**Fragmented**

In this design, a table is divided into two or more pieces referred to as fragments or partitions, and each fragment can be stored at different sites. This considers the fact that it seldom happens that all data stored in a table is required at a given site. Moreover, fragmentation increases parallelism and provides better disaster recovery. Here, there is only one copy of each fragment in the system, i.e. no redundant data.

**Mixed Distribution**

This is a combination of fragmentation and partial replications. Here, the tables are initially fragmented in any form (horizontal or vertical), and then these fragments are partially replicated across the different sites according to the frequency of accessing the fragments.