



# MODUL II

## KONSEP & MANAJEMEN BASIS DATA

# LINGKUNGAN & ARSITEKTUR

# BASIS DATA

---

*Disarikan dari:* Connolly, Thomas & Carolyn Begg, 2015, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6<sup>th</sup> ed, Chapter 1-3

Prodi Informatika  
Universitas Sanata Dharma

# Capaian Pembelajaran Pokok Bahasan

- Mahasiswa memahami konsep, struktur, dan arsitektur sistem basisdata

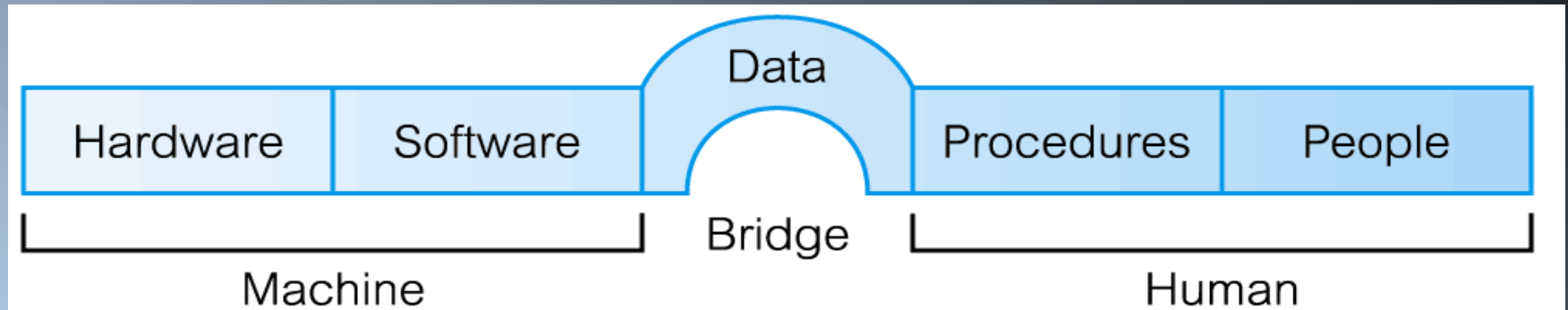
# Basis Data

- ❑ Basis data:
  - kumpulan data yang secara logis terkait satu sama lain dan digunakan bersama oleh banyak pemakai, serta dirancang untuk memenuhi kebutuhan informasi banyak pemakai dalam suatu organisasi.
- ❑ Data yang terkait secara logis berisi informasi dalam organisasi/domain masalah yang terdiri atas entitas, atribut, dan hubungannya
- ❑ Selain data, terdapat juga metadata yang berisi deskripsi data.

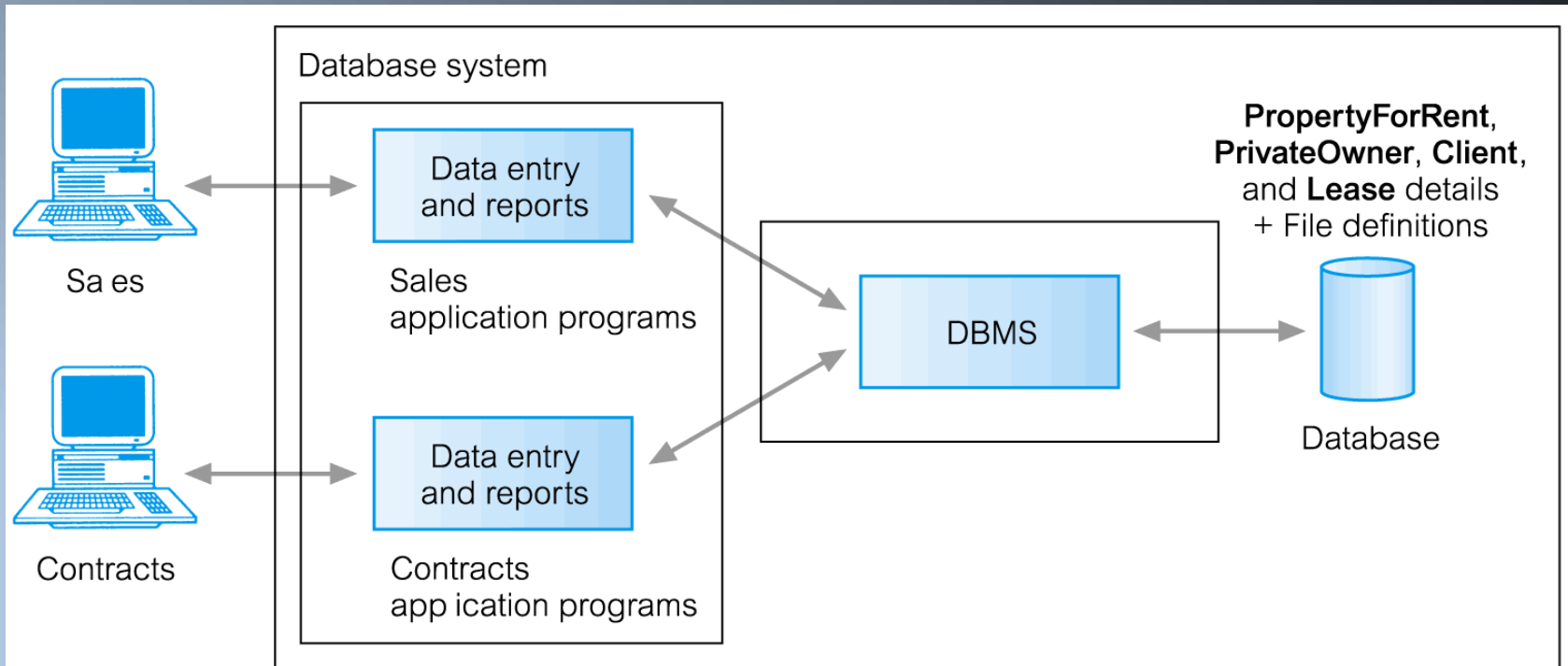
# *Database Management System (DBMS)*

- ❑ Sistem manajemen basis data (*Database Management System/DBMS*):
  - sistem perangkat lunak komersial yang digunakan untuk membuat, merawat dan menyediakan akses terkendali atas basisdata dan repository. Contoh: Oracle, Foxpro, dsb
- ❑ Program aplikasi (basisdata):
  - program komputer yang berinteraksi dengan basisdata dengan melakukan request (perintah SQL) ke DBMS

# Komponen Lingkungan DBMS



# Ilustrasi Database Management System



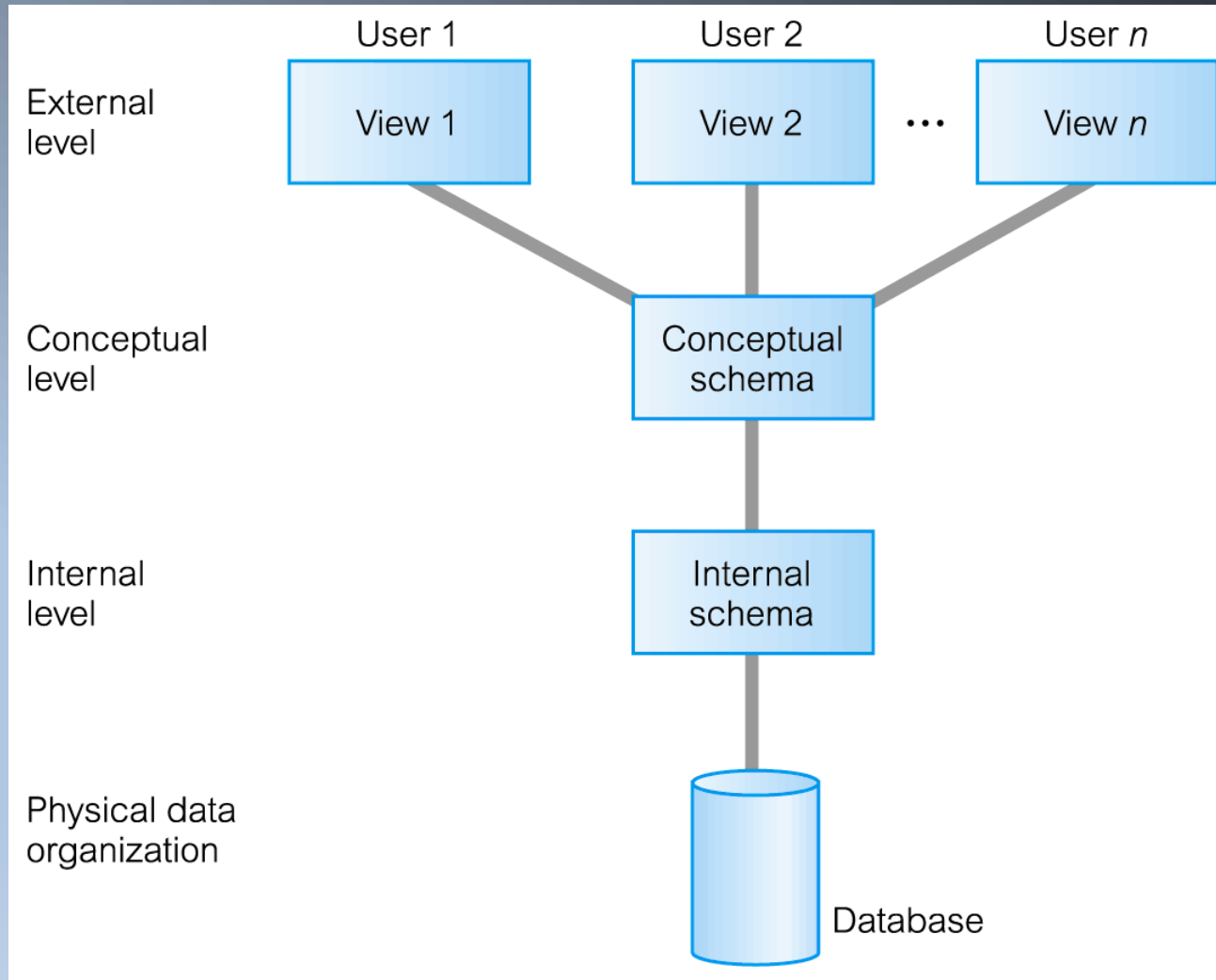
**PropertyForRent** (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

**PrivateOwner** (ownerNo, fName, lName, address, telNo)

**Client** (clientNo, fName, lName, address, telNo, prefType, maxRent)

**Lease** (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

# Arsitektur Basis Data (Tiga-Level ANSI-SPARC)



# Arsitektur Basis Data (Tiga-Level ANSI-SPARC)

- ❑ Level eksternal (*external level*)
  - View database milik pengguna
  - Menggambarkan sebagian basisdata yang relevan dengan kebutuhan pengguna
- ❑ Level konseptual (*conceptual level*)
  - Kumpulan view dari basisdata
  - Menggambarkan data apa saja yang tersimpan dalam basisdata dan hubungan antar data tersebut.
- ❑ Level internal (*Internal Level*)
  - Representasi fisik basisdata dalam komputer
  - Menggambarkan bagaimana data disimpan dalam basisdata



# Tujuan Arsitektur Tiga Level

- ❑ Semua pengguna dapat mengakses data yang sama
- ❑ View dari seorang pengguna tidak terpengaruh perubahan view dari pengguna yang lain
- ❑ Pengguna tidak perlu tahu secara detil simpanan fisik basisdata
- ❑ DBA (Data Base Administrator) harus dapat mengubah struktur simpanan basisdata tanpa mempengaruhi view pengguna.
- ❑ Struktur Internal basisdata tidak boleh terpengaruh oleh aspek fisikal simpanan data.
- ❑ DBA harus dapat mengubah struktur konseptual basidata tanpa mempengaruhi semua pengguna.

# Perbedaan Tiga Level Arsitektur ANSI-SPARC

External view 1

sNo	fName	lName	age	salary
-----	-------	-------	-----	--------

External view 2

staffNo	lName	branchNo
---------	-------	----------

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo
---------	-------	-------	-----	--------	----------

Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOf Birth;  
    float salary;  
    struct STAFF *next;           /* pointer to next Staff record */  
};  
index staffNo; index branchNo; /* define indexes for staff */
```

# Independensi Data

## ❑ Independensi data logikal

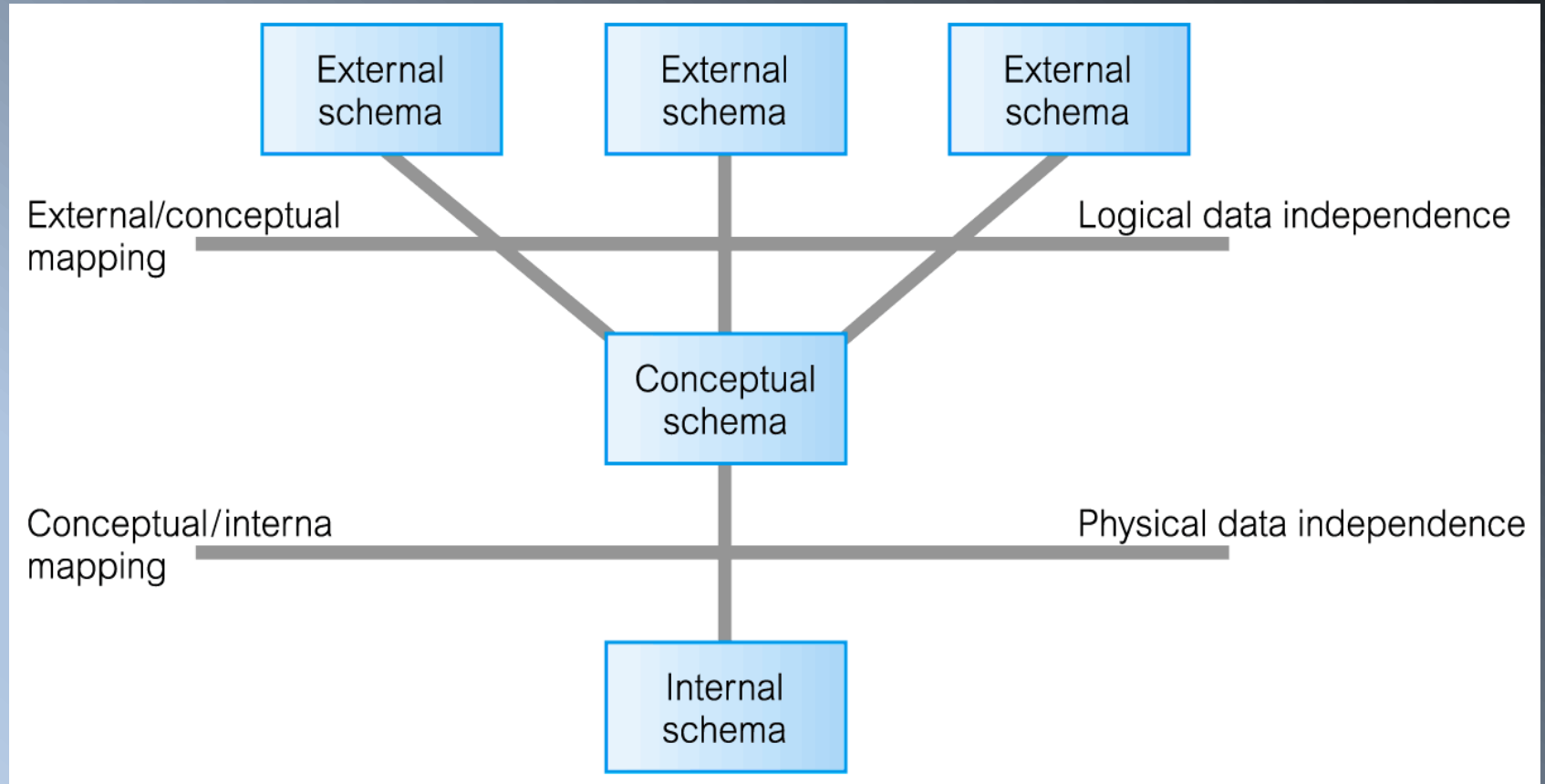
- kekebalan skema eksternal terhadap perubahan yang dilakukan di skema konseptual
- Jika skema konseptual berubah (misal karena penambahan/penghapusan entitas) , maka tidak boleh mengakibatkan perubahan skema eksternal atau perubahan program aplikasi

# Independensi Data

## ❑ Independensi data fisikal

- kekebalan perubahan skema konseptual terhadap perubahan skema internal
- Jika skema internal berubah (misal karena perubahan organisasi file, perubahan struktur simpanan data/alat penyimpanan), maka tidak boleh mempengaruhi skema konseptual/skema eksternal

# Independensi Data dan Arsitektur Tiga-Level ANSI-SPARC



# Peran dalam Lingkungan Basisdata

## ❑ *Data Administrator (DA):*

- Bertanggungjawab atas manajemen data (perencanaan, pengembangan, perawatan standar, prosedur, kebijakan, dan rancangan logikal/konseptual)

## ❑ *Database Administrator (DBA):*

- Bertanggungjawab atas realisasi fisik basisdata (rancangan fisik, implementasi, keamanan, integritas, perawatan sistem operasional, kinerja aplikasi)

## ❑ *Database Designers (Logical dan Physical):*

- Bertanggungjawab atas rancangan logikal dan rancangan fisik basisdata

# Peran dalam Lingkungan Basisdata

## ❑ *Application Programmers:*

- Bertanggungjawab atas pembuatan program aplikasi yang memanfaatkan basisdata yang ada.

## ❑ *End Users (awam dan lanjut):*

- Pengguna basisdata
- Awam: tidak menyadari adanya DBMS, mengakses BD melalui program aplikasi
- Lanjut: mempunyai kemampuan untuk melakukan kueri menggunakan SQL atau membuat program aplikasi sendiri

# Database Language

## ❑ *Data Definition Language (DDL)*

- Memungkinkan DBA atau pengguna untuk mendeskripsikan dan memberi nama entitas, atribut, dan hubungan-hubungan yang diperlukan dalam suatu aplikasi
- Memuat juga informasi tentang integritas dan kendala keamanan data (*security constraints*) yang relevan

## ❑ *Data Manipulation Language (DML)*

- Menyediakan operasi-operasi dasar untuk memanipulasi data yang tersimpan dalam basisdata

## ❑ *Fourth Generation Languages (4GLs):*

- **4GL** adl Bahasa pemrograman yang lebih dekat ke Bahasa manusia daripada Bahasa pemrograman tingkat tinggi spt Java atau C++ .
- SQL adalah salah satu contoh 4GL



# Tipe DML

## ❑ *Procedural DML*

- Memungkinkan pengguna untuk memberitahu sistem **bagaimana cara memanipulasi data**
- Biasanya ditanam (embedded) di dalam Bahasa pemrograman tingkat tinggi
- Model data network & hierarchical biasanya termasuk procedural DML.

## ❑ *Non-Procedural DML*

- Memungkinkan pengguna untuk menyatakan **data apa yang dibutuhkan**, dan **bukan bagaimana cara** membaca data tersebut
- Contoh: SQL, QBE

# Model Data (*Data Model*)

- ❑ Koleksi terintegrasi dari konsep tentang deskripsi data, hubungan antar data, dan kendala terhadap data dalam suatu organisasi
  
- ❑ Model data memuat :
  - bagian struktural
  - Bagian manipulatif
  - Mungkin juga memuat sekumpulan aturan integritas

# Model Data

## ❑ Tujuan

- Untuk merepresentasikan data dengan cara yang dapat dimengerti

## ❑ Kategori model data meliputi :

- Berbasis obyek (*Object-based*)
- Berbasis record (*Record-based*)
- Fisikal (*Physical*)

# Model Data

## ❑ Model Data Berbasis Obyek

- Hubungan antar entitas (*Entity-Relationship*)
- Semantik
- Fungsional
- Berorientasi Obyek (*Object-Oriented*).

## ❑ Model Data Berbasis Record

- Model Data Relasional
- Model Data Network
- Model Data Hirarkikal

## ❑ Model Data Fisikal

# Sejarah Sistem Basisdata

- ❑ First-generation
  - Hierarchical dan Network
- ❑ Second generation
  - Relasional
- ❑ Third generation
  - Object-Relational
  - Object-Oriented

# Model Data Relasional

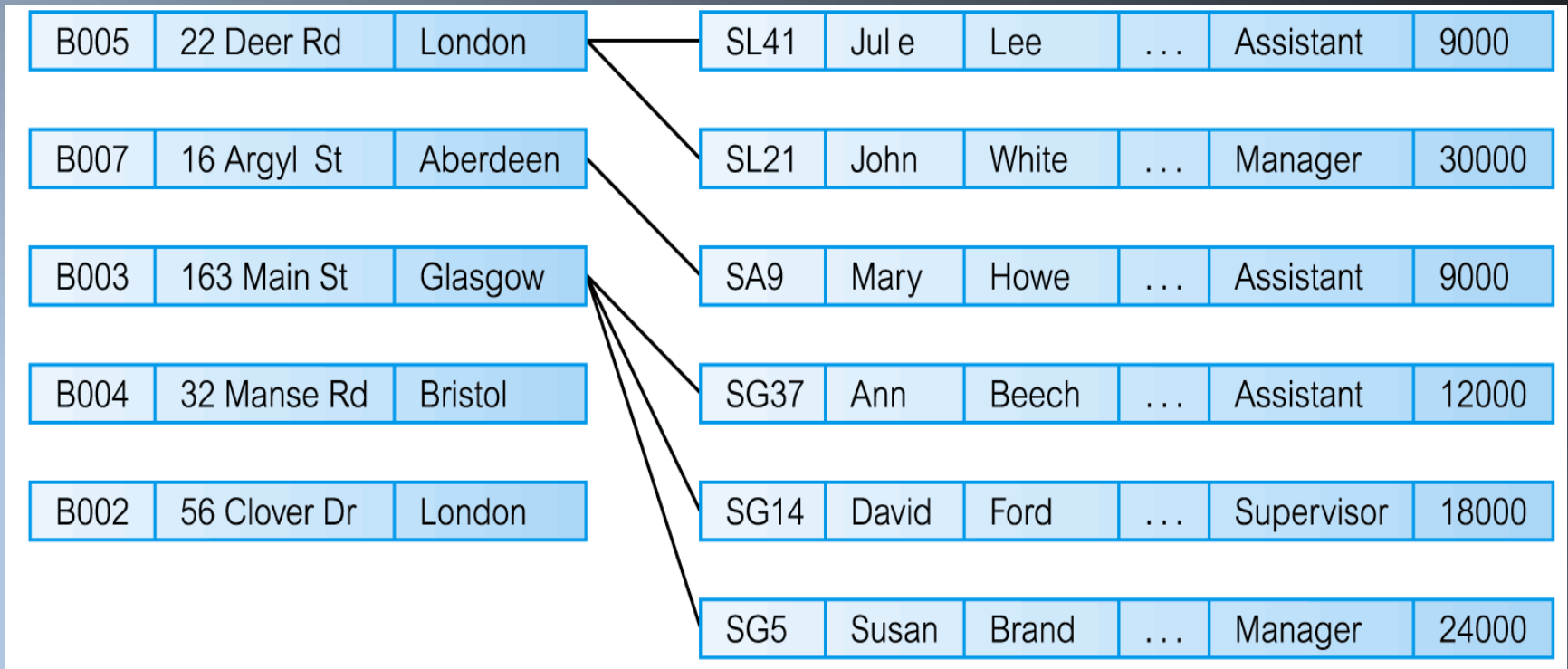
**Branch**

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

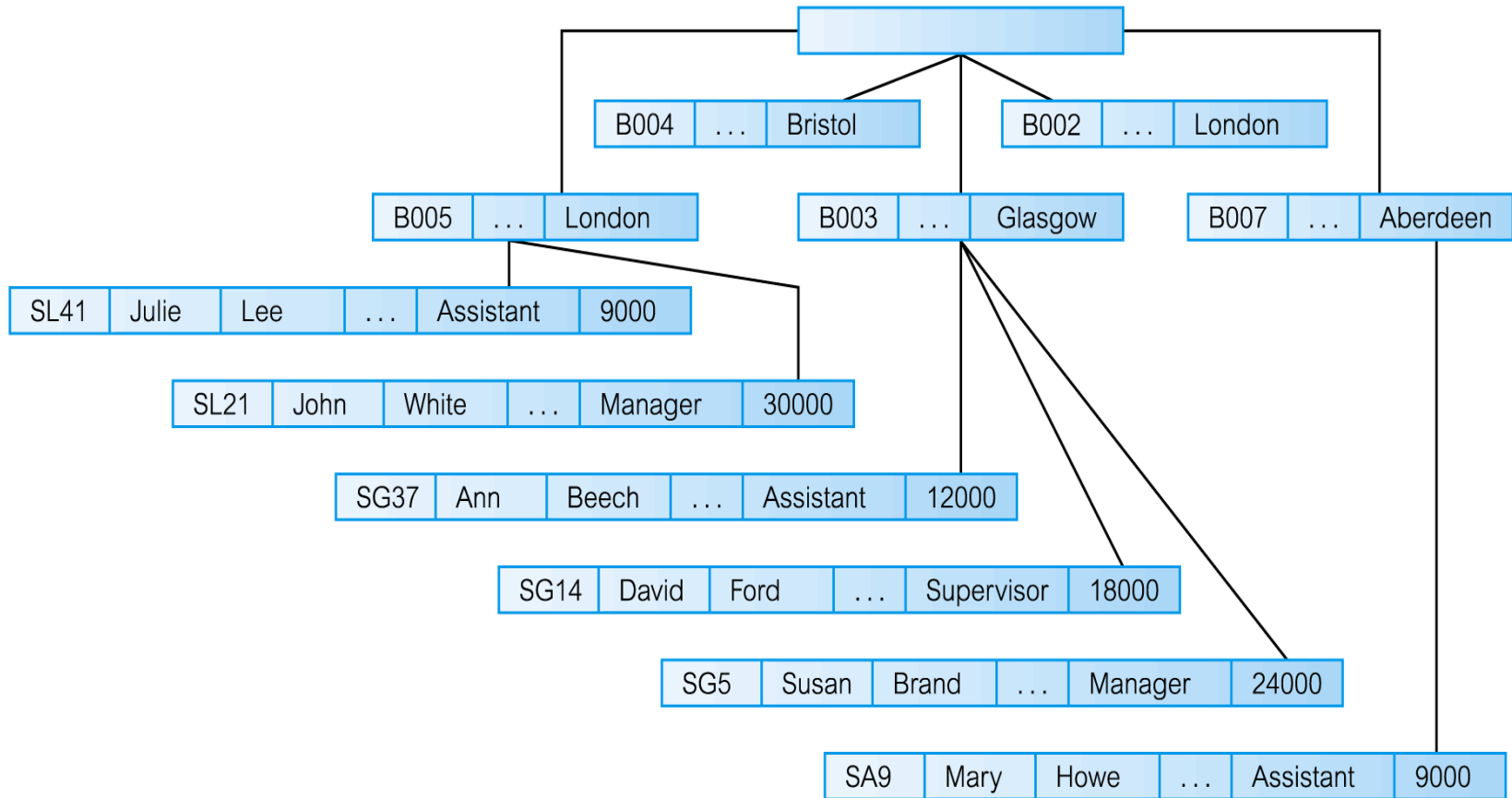
**Staff**

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

# Model Data Network



# Model Data Hirarkikal

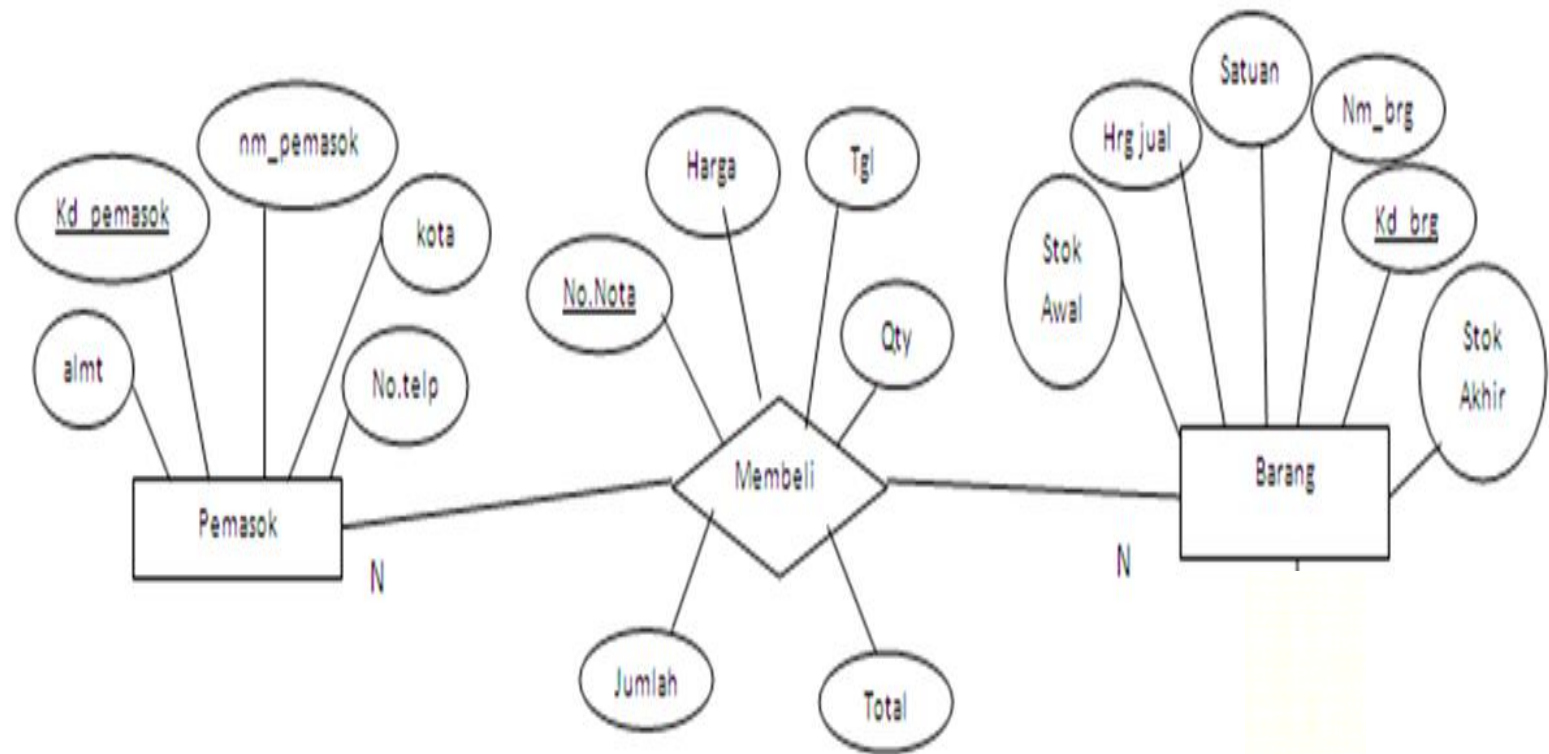




# Pemodelan Konseptual

- ❑ Skema konseptual adalah inti dari suatu sistem yang mendukung seluruh view pengguna .
- ❑ Harus merepresentasikan secara lengkap dan akurat kebutuhan data dari organisasi
- ❑ Pemodelan konseptual adalah proses pembuatan model penggunaan informasi yang independen terhadap detail implementasi
- ❑ Hasil dari pemodelan konseptual adalah model data konseptual
- ❑ **Contoh:**
  - Model ER SPPK Penghitungan Biaya Diklat

# Pemodelan Konseptual

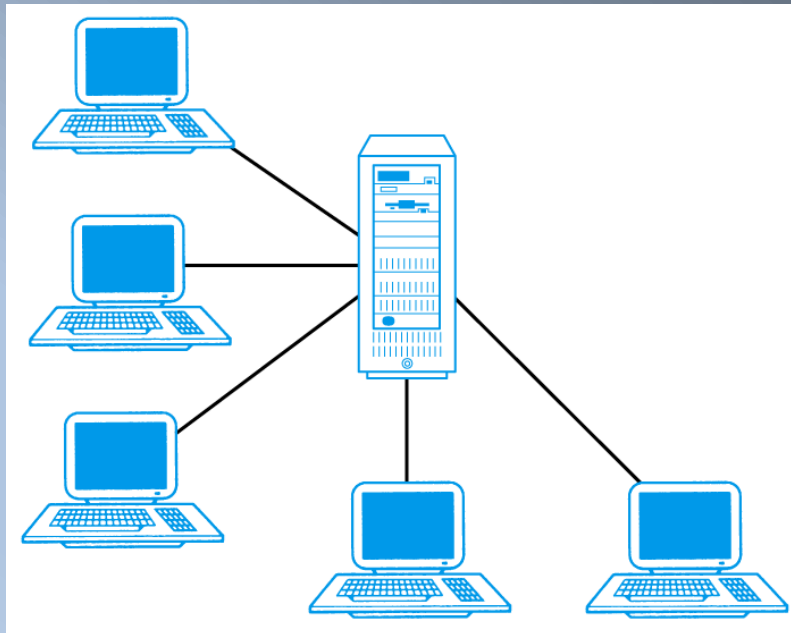


# Arsitektur *Multi-User DBMS*

- ❑ *Teleprocessing*
- ❑ *File-server*
- ❑ *Client-server*
- ❑ *Web Service*
- ❑ *SOA*
- ❑ *Distributed DBMS*
- ❑ *Cloud Computing*

# Teleprocessing

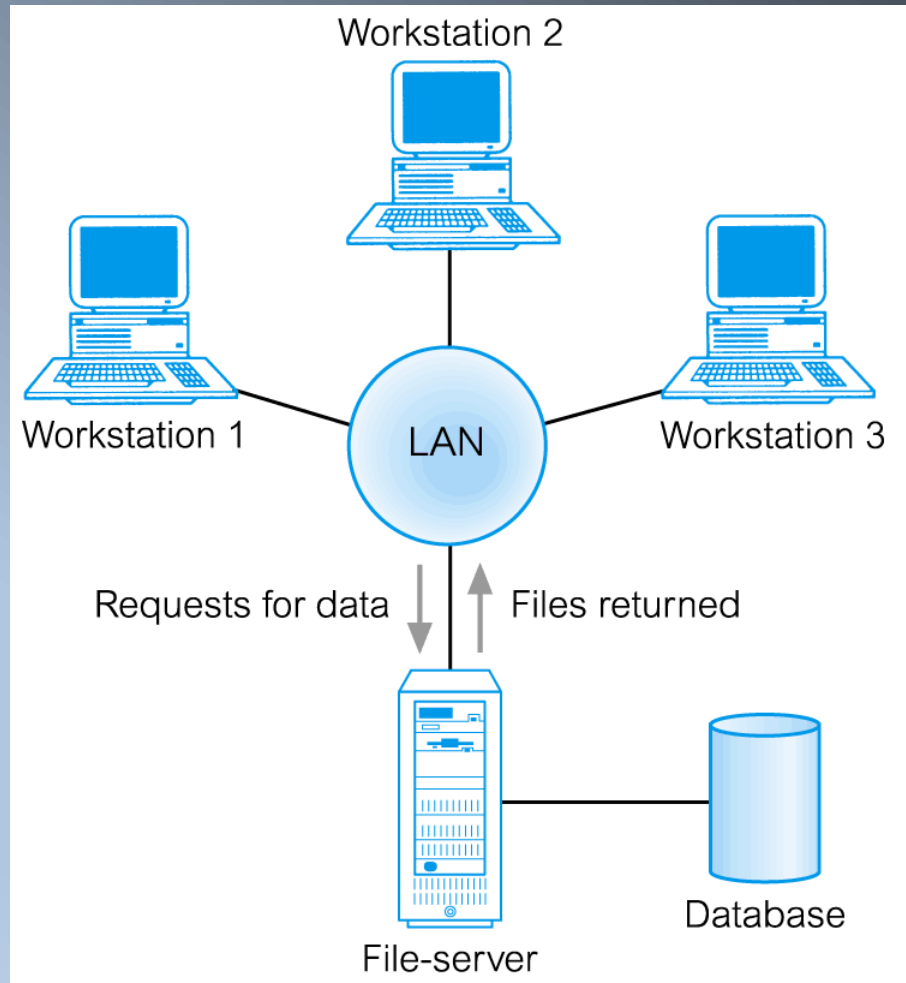
- ❑ Arsitektur tradisional
- ❑ Satu mainframe dengan sejumlah terminal yang terhubung dengannya
- ❑ Trend saat ini adalah *downsizing*



# *File-Server*

- ❑ File-server terhubung ke beberapa *workstations* dalam suatu jaringan
- ❑ Basisdata tersimpan dalam file-server.
- ❑ DBMS dan aplikasi berjalan dalam setiap workstation.
- ❑ Kelemahan:
  - Mengakibatkan padatnya lalu lintas jaringan
  - Terdapat duplikat (copy) DBMS dalam setiap workstation.
  - Kontrol terhadap konkurensi, pemulihan (*recovery*) dan integritas data lebih kompleks

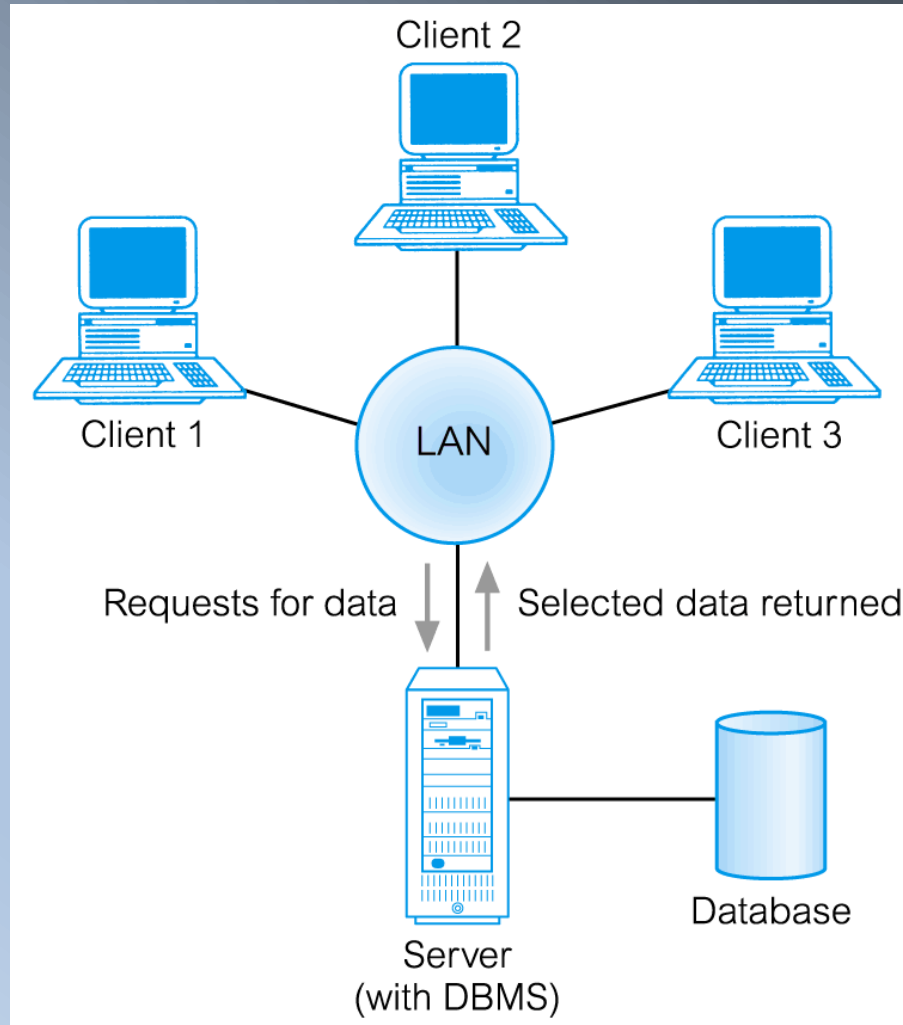
# Arsitektur File-Server



# ***Traditional Two-Tier Client-Server***

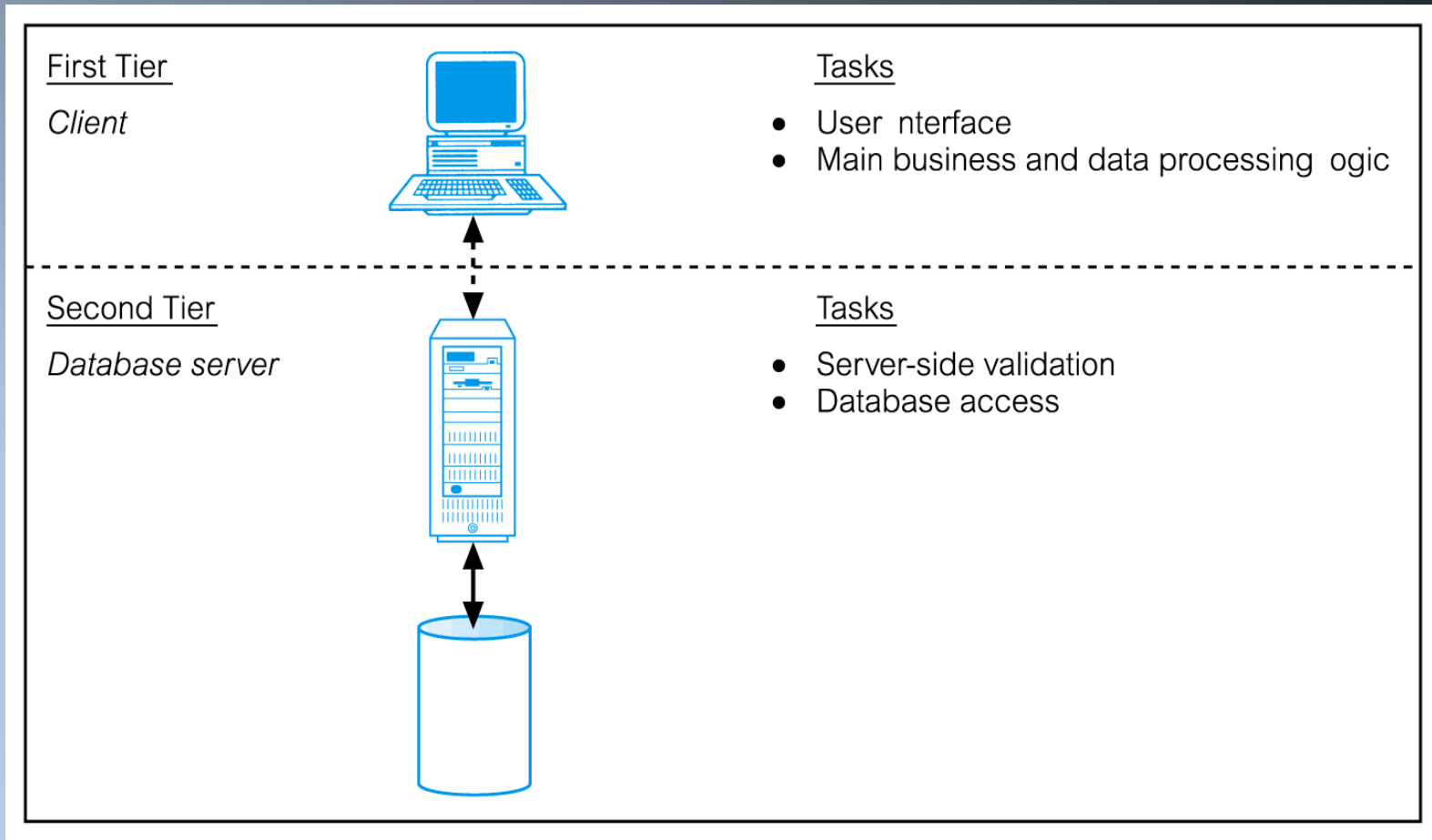
- ❑ Client (tier 1) mengelola antarmuka pengguna (*user interface*) dan menjalankan aplikasi
- ❑ Server (tier 2) menyimpan basisdata dan DBMS.
- ❑ Kelebihan:
  - Akses yang lebih luas terhadap basisdata yang ada
  - Kinerja meningkat
  - Dapat mengurangi biaya hardware
  - Mengurangi biaya komunikasi
  - Konsistensi meningkat

# *Traditional Two-Tier Client-Server*





# Traditional Two-Tier Client-Server



# Three-Tier Client-Server

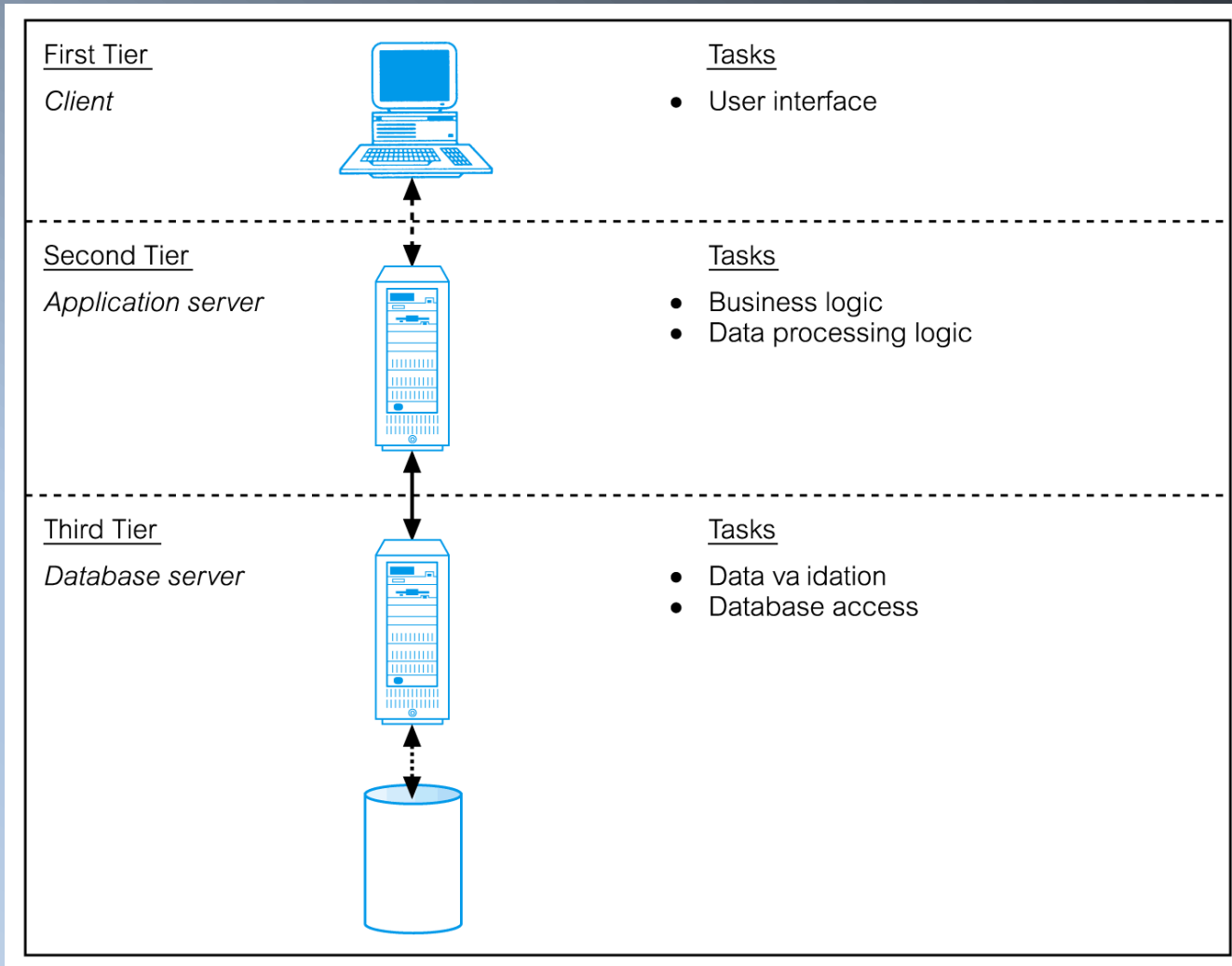
- ❑ Dalam arsitektur 2-tier, sisi klien (*Client side*) menghadapi 2 masalah yang menghambat aplikasi berskala enterprise (mencakup seluruh organisasi yang cukup besar)
  - *'Fat' client*: membutuhkan sumberdaya yang memadai bagi komputer sisi klien agar program dapat berjalan efektif
  - Membutuhkan biaya overhead yang signifikan di sisi klien .
- ❑ Tahun 1995, diusulkan konsep *three layers* yang masing-masing bisa berjalan pada platform yang berbeda

# Three-Tier Client-Server

## ❑ Keuntungan:

- *'Thin' client*: membutuhkan biaya hardware yang lebih murah
- Perawatan aplikasi terpusat
- Mudah untuk memodifikasi atau mengubah suatu tier tanpa mempengaruhi yang lain
- Memisahkan logika bisnis dari fungsi basisdata dan memudahkan implementasi penyeimbangan beban kerja (*load balancing*)
- Lebih mudah dipetakan ke lingkungan web

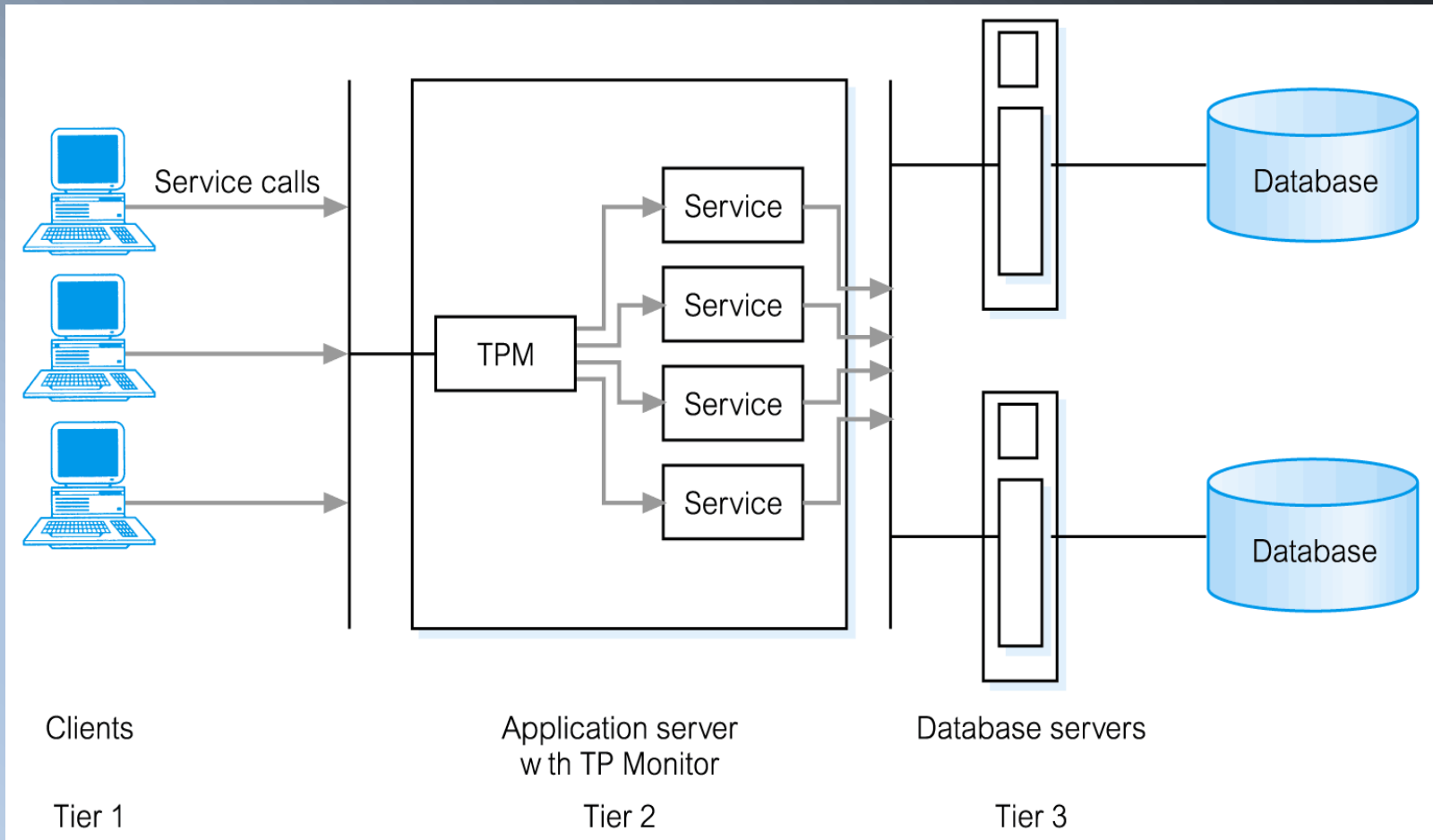
# Three-Tier Client-Server



# ***Transaction Processing Monitors***

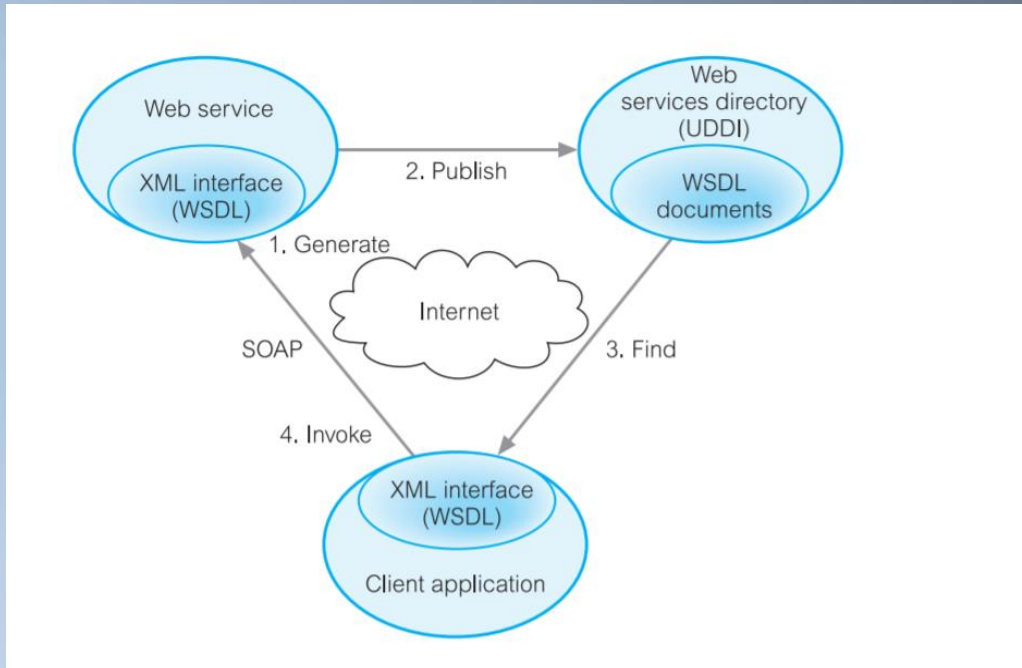
- ❑ Program yang mengendalikan transfer data antara klien dan server untuk menjamin lingkungan yang konsisten, terutama untuk *Online Transaction Processing (OLTP)*.

# TPM sebagai *middle tier* dari 3-tier client-server



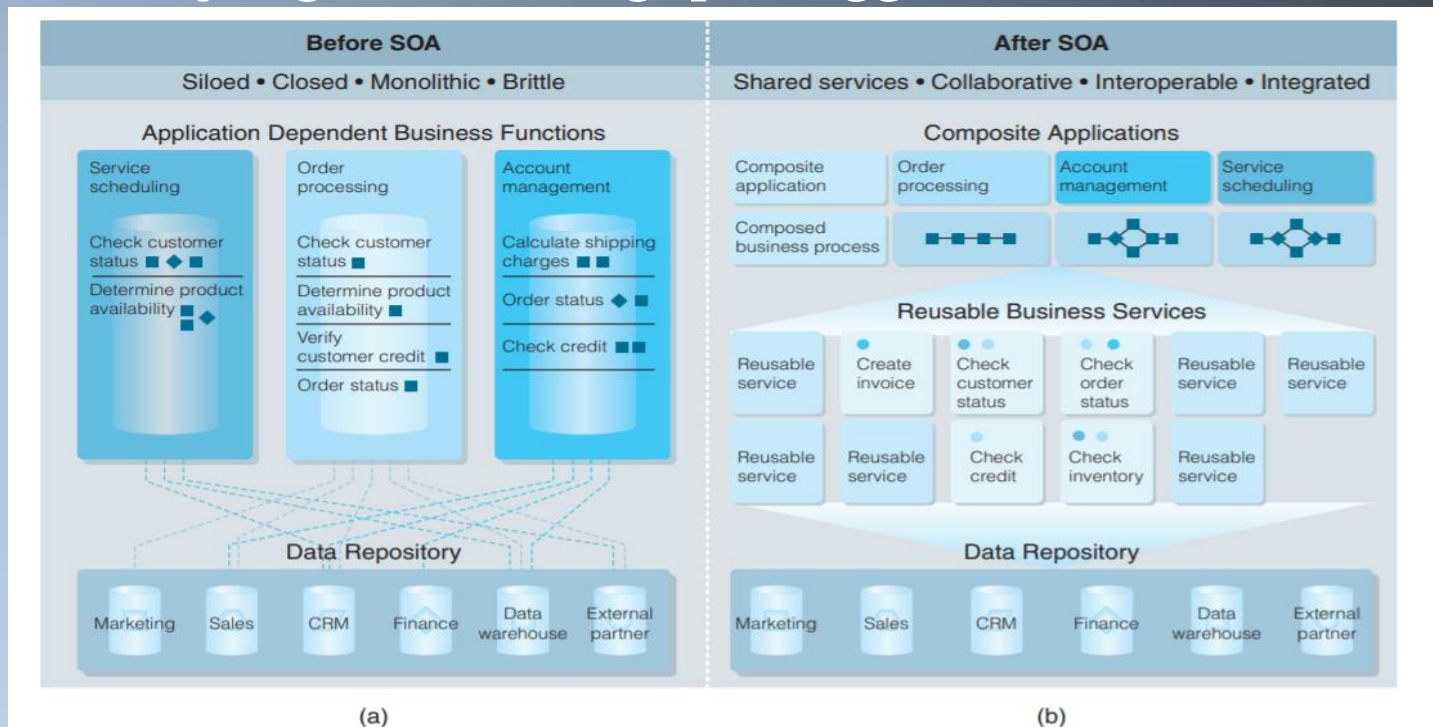
# Web Services

- Sebuah sistem perangkat lunak yg dirancang untuk mendukung interoperabilitas interaksi mesin ke mesin melalui jaringan internet.
- Kunci: menggunakan teknologi & standar yang diterima banyak pihak (contoh: XML, WSDL, SOAP, UDDI)



# Service-Oriented Architectures (SOA)

- Suatu arsitektur software yang berpusat pada bisnis (business-centric) untuk membangun aplikasi yang mengimplementasikan proses bisnis sebagai kumpulan servis yang relevan bagi pelanggan. *Sumber: Connolly & Begg (2015)*

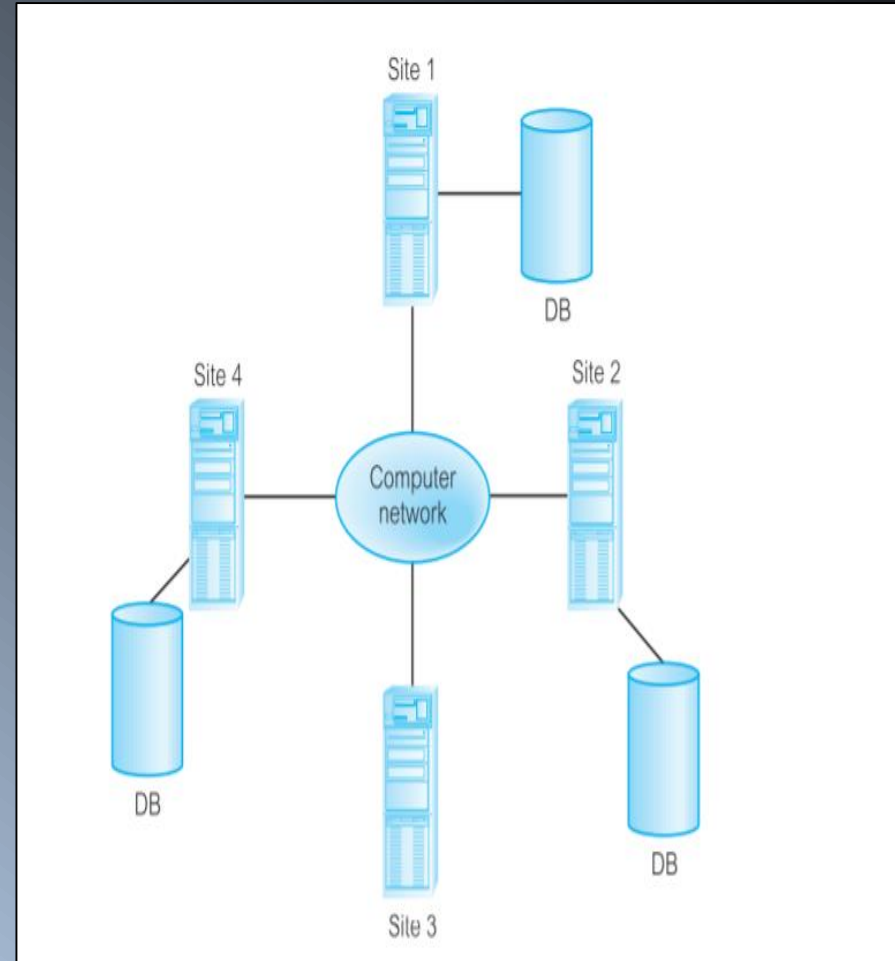


**Figure 3.10** (a) Traditional IT architecture for three business processes; (b) service-oriented architecture that splits the processes into a number of reusable services.



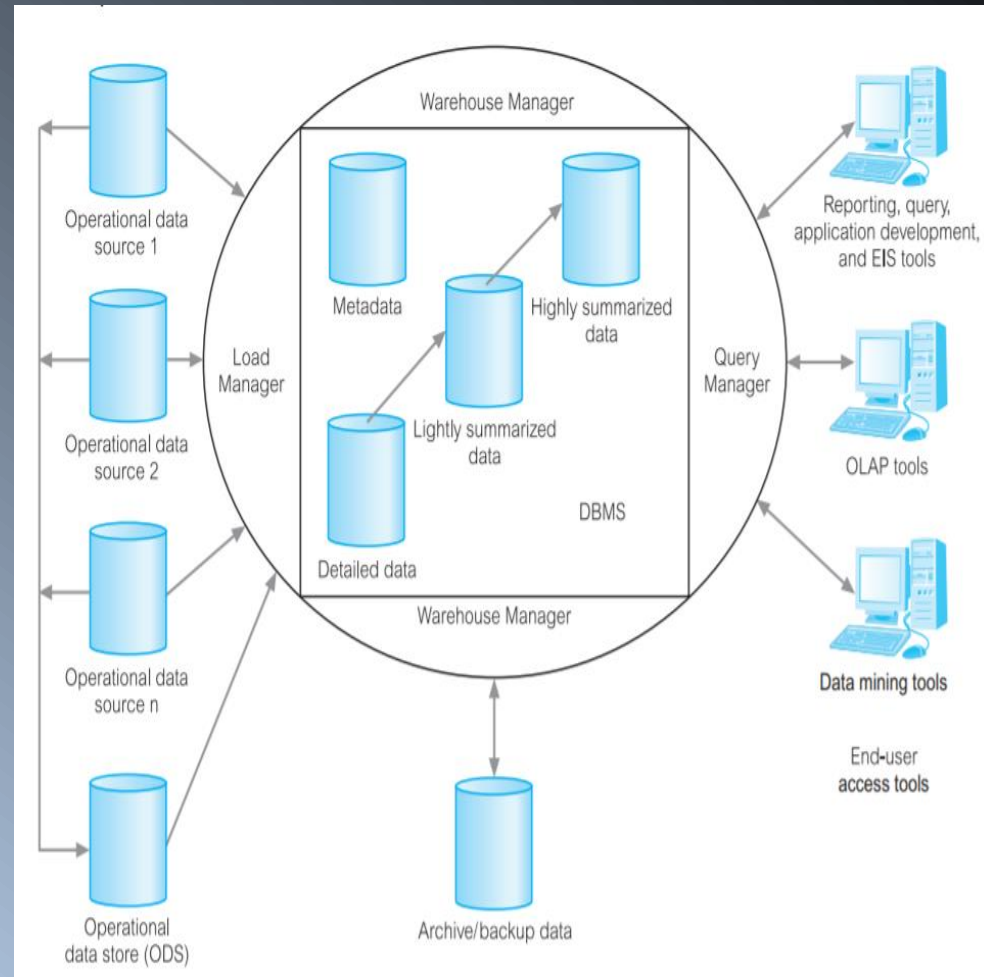
# Distributed DBMS

- **Distributed database:** sekumpulan data bersama (beserta deskripsinya) yang secara logis berkaitan satu sama lain, dan secara fisik terdistribusi pada suatu jaringan computer
- **Distributed DBMS:** sistem perangkat lunak yang memfasilitasi pengelolaan distributed database dan membuatnya menjadi transparan bagi pengguna



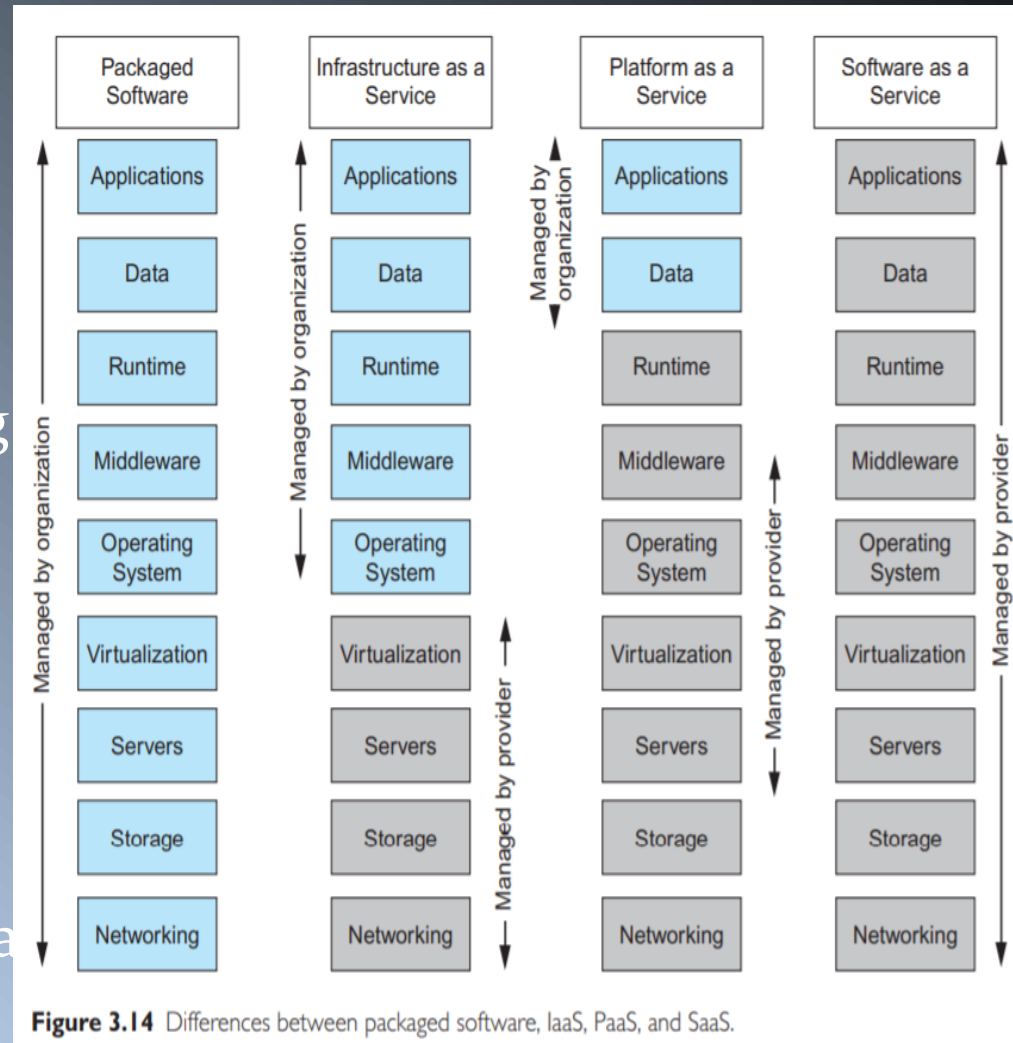
# Data Warehouse

- Sekumpulan **view data** korporasi yang terintegrasi dan berasal dari berbagai sumber data operasional yang terpisah-pisah **serta berbagai piranti akses** bagi **end-user** yang mampu mendukung query yang simple maupun yang kompleks untuk membantu pengambilan keputusan



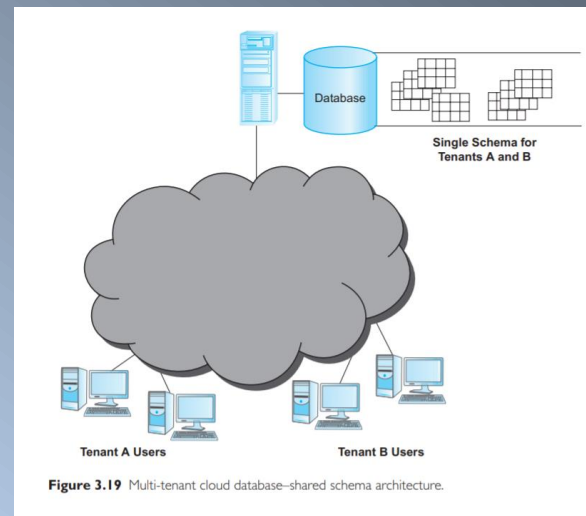
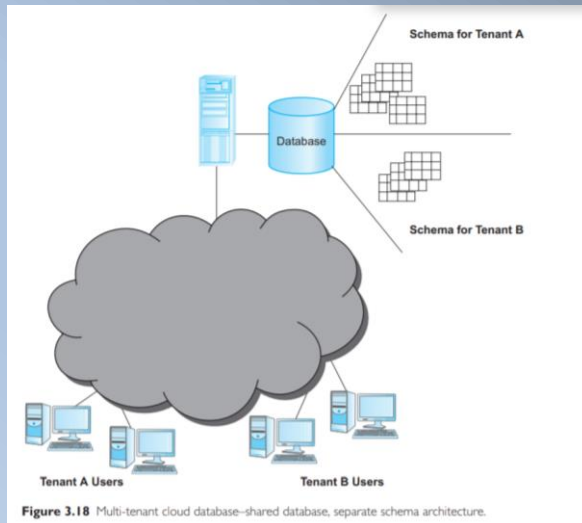
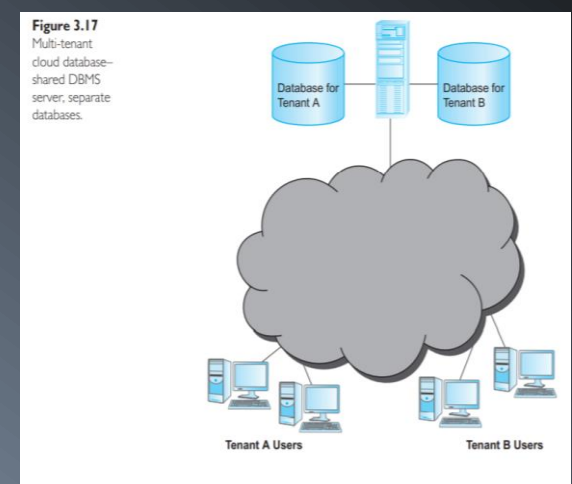
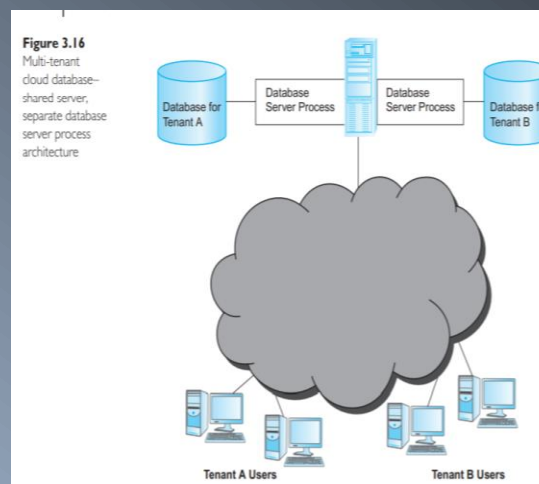
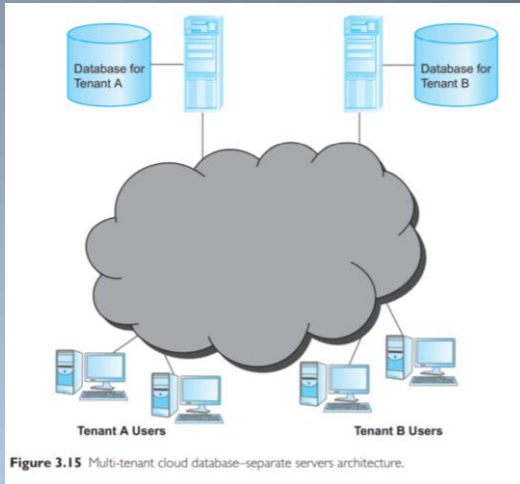
# Cloud Computing

- Suatu model yang memfasilitasi akses melalui jaringan komputer dari manapun secara nyaman, berdasar kebutuhan terhadap sumberdaya komputing yang dapat dikonfigurasi (misal networks, servers, storage, applications, dan services) dan disediakan secara cepat dengan manajemen pengelolaan yang minimum atau melalui layanan penyedia jasa yang ditetapkan NIST



Sumber: Connolly & Begg (2015)

# Berbagai Arsitektur Cloud Computing



# Keuntungan DBMS

- ❑ Mengendalikan redundansi data
- ❑ Menjamin konsistensi data
- ❑ Menghasilkan informasi yang lebih banyak dari jumlah data yang sama
- ❑ Memungkinkan pemakaian data secara bersama (*data sharing*)
- ❑ Meningkatkan integritas data
- ❑ Meningkatkan keamanan data
- ❑ Mendorong standarisasi
- ❑ Bisa lebih hemat dibandingkan jika menggunakan file-based system

# Keuntungan DBMS

- ❑ Menyeimbangkan kebutuhan yang bertentangan
- ❑ Meningkatkan aksesibilitas dan respon data
- ❑ Meningkatkan produktifitas
- ❑ Meningkatkan perawatan melalui independensi data
- ❑ Meningkatkan konkurensi (keserempakan)
- ❑ Meningkatkan layanan backup dan recovery

# Kelemahan DBMS

- ❑ Kompleksitas
- ❑ Ukuran
- ❑ Biaya DBMS
- ❑ Biaya tambahan untuk hardware
- ❑ Biaya konversi
- ❑ Kinerja (yang mungkin lebih lambat dibanding file-based system)
- ❑ Memberikan dampak kegagalan yang lebih besar



# *Wisdom of the Day*

Lakukanlah segala sesuatunya  
dengan gembira

Maka hasilnya akan  
menggembirakan hati...

