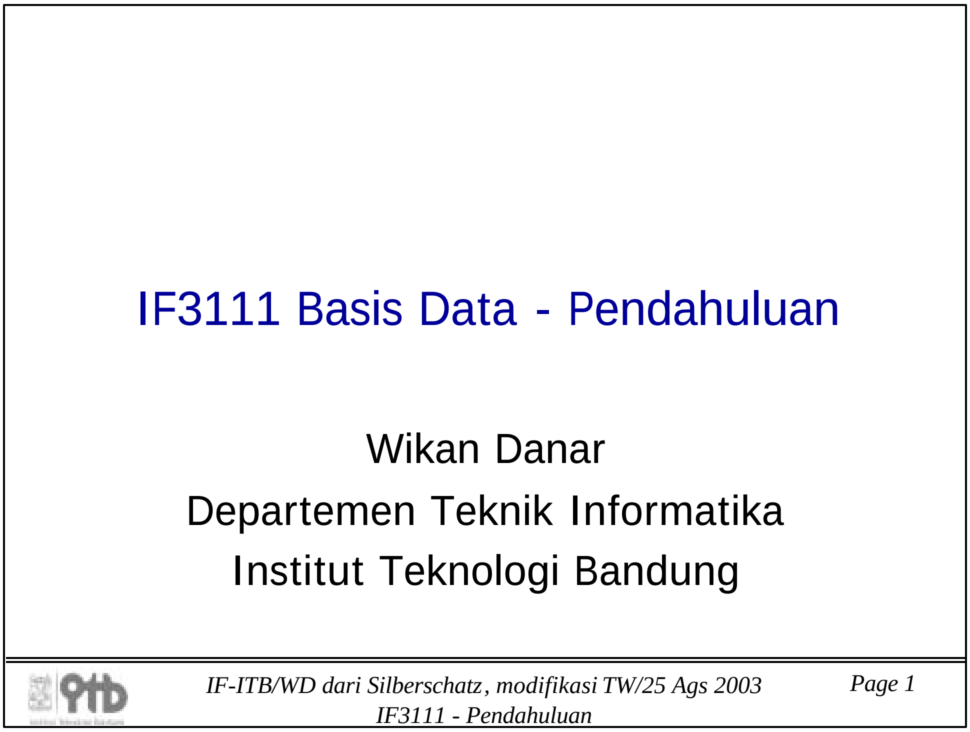


IF3111 Basis Data - Pendahuluan

Wikan Danar
Departemen Teknik Informatika
Institut Teknologi Bandung

Tujuan Instruksional Umum:

Pada kuliah ini diberikan pengetahuan umum tentang basis data dan sistem basis data, dan ditumbuhkan kemampuan untuk:

1. Membedakan antara kedua terminologi tersebut.
2. Membedakan data dari ketiga sudut pandang (level abstraksi).
3. Mengkategorisasi pemakai sistem basis data.
4. Menjelaskan secara umum komponen-komponen dari suatu sistem basis data.

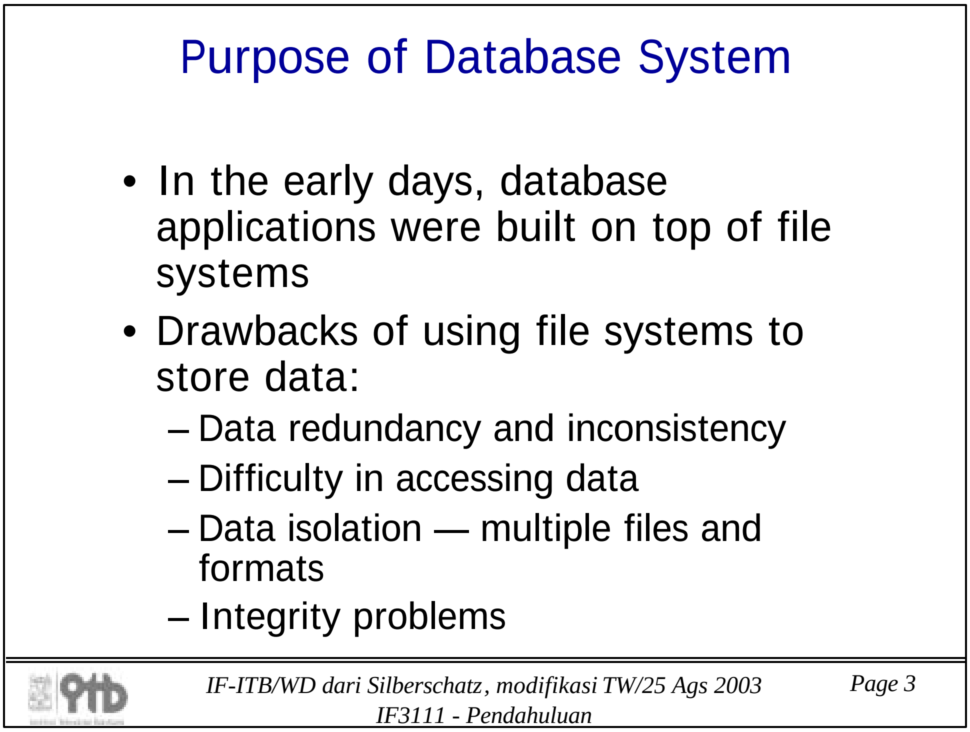
Referensi:

[DAT00] Date, C.J., *An Introduction to Database System*, 7th edition, Addison Wesley, 2000

[SIL99] Silberschatz, A., Korth, H. F., Sudarshan, S., *Database System Concepts*, 3rd edition, McGraw-Hill, 1999

Database Management System (DBMS)

- Collection of interrelated data
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that is both *convenient* and *efficient* to use.
- Database Applications:
  - Banking: all transactions
  - Universities: registration, grades
  - Sales: customers, products, purchases
- Databases touch all aspects of our lives



Purpose of Database System

- In the early days, database applications were built on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
  - Difficulty in accessing data
  - Data isolation — multiple files and formats
  - Integrity problems

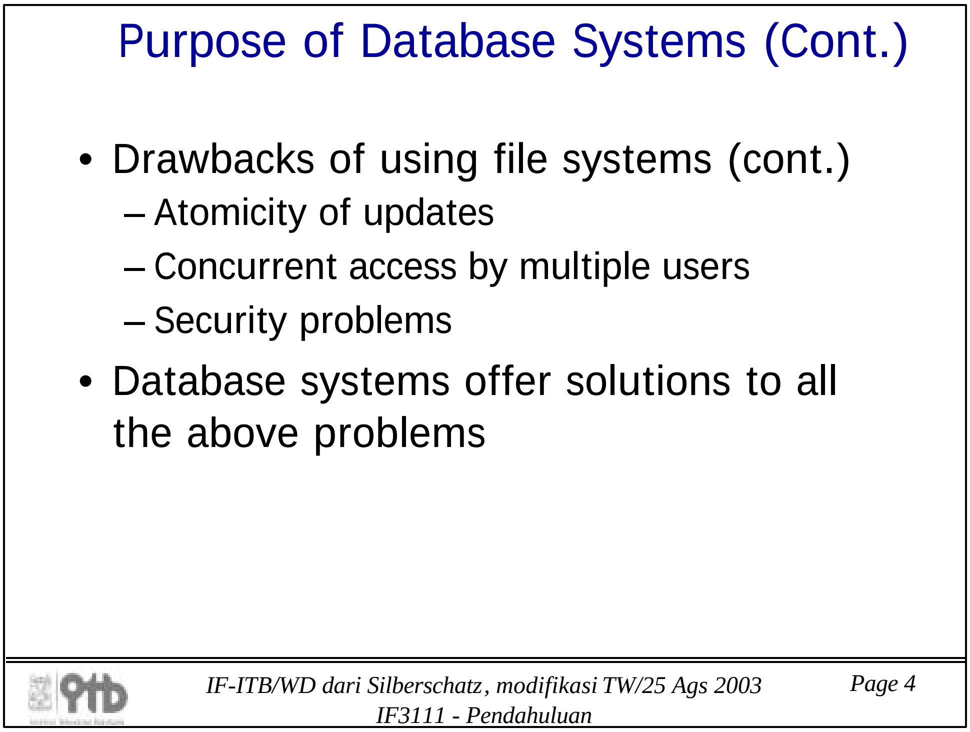
Contoh data redundansi dan inkonsistensi adalah file format yang beragam, duplikasi informasi di file yang berbeda

Contoh kesulitan dalam pengaksesan data : diperlukan pembuatan program baru setiap pelaksanaan tugas baru

Contoh permasalahan integritas :

•Integrity constraints (contoh : saldo rekening > 0) menjadi bagian dari kode program

•Kesulitan dalam penambahan kendala baru atau pengubahan kendala yang sudah ada



Purpose of Database Systems (Cont.)

- Drawbacks of using file systems (cont.)
  - Atomicity of updates
  - Concurrent access by multiple users
  - Security problems
- Database systems offer solutions to all the above problems

Atomicity of updates

•Failures may leave database in an inconsistent state with partial updates carried out

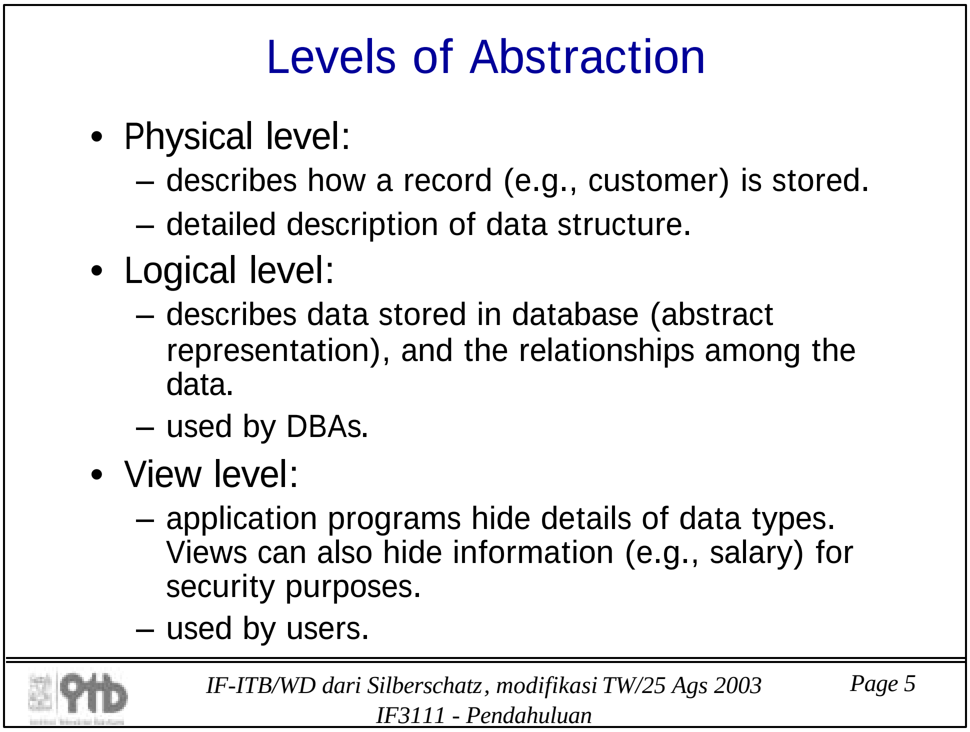
•E.g. transfer of funds from one account to another should either complete or not happen at all

Concurrent access by multiple users

•Concurrent accessed needed for performance

•Uncontrolled concurrent accesses can lead to inconsistencies

> E.g. two people reading a balance and updating it at the same time



Levels of Abstraction

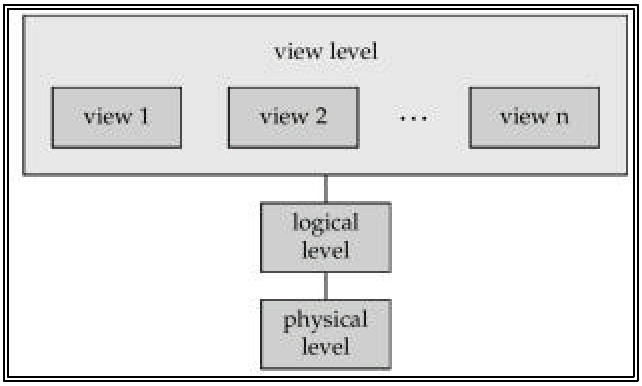
- Physical level:
  - – describes how a record (e.g., customer) is stored.
  - – detailed description of data structure.
- Logical level:
  - – describes data stored in database (abstract representation), and the relationships among the data.
  - – used by DBAs.
- View level:
  - – application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.
  - – used by users.

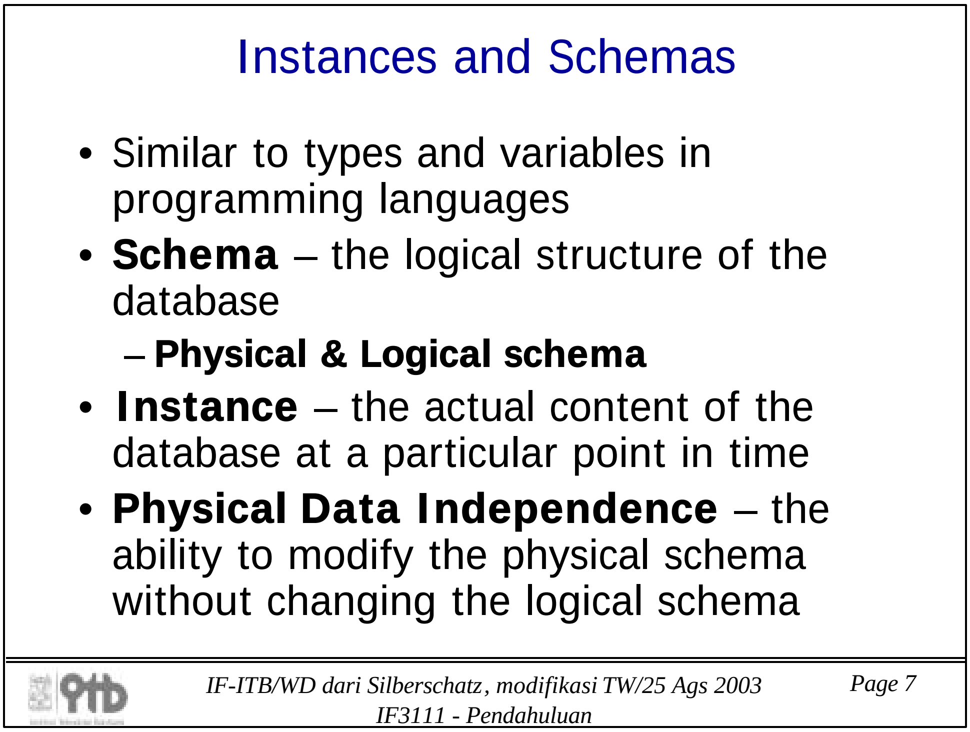
Contoh logical level :

```
            type customer = record
                                       name : string;
                                       street : string;
                                       city : integer;
                      end;
```

View of Data

An architecture for a database system (ANSI/SPARC Architecture)





Instances and Schemas

- Similar to types and variables in programming languages
- **Schema** – the logical structure of the database
  - **Physical & Logical schema**
- **Instance** – the actual content of the database at a particular point in time
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema

IF-ITB/WD dari Silberschatz, modifikasi TW/25 Ags 2003 *Page 7*
IF3111 - Pendahuluan

Schema :

•e.g., the database consists of information about a set of customers and accounts and the relationship between them)

•Analogous to type information of a variable in a program

•**Physical schema**: database design at the physical level

•**Logical schema**: database design at the logical level

Instance : Analogous to the value of a variable

Physical Data Independence :

•Applications depend on the logical schema

•In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Skema memiliki frekuensi update yang rendah, sedangkan untuk instans sangat tinggi.

Umumnya, sebuah basis data akan memiliki satu skema fisik, satu skema konseptual, dan beberapa subskema.

Data Definition Language (DDL)

- Specification notation for defining the database schema
  - E.g.

```
create table account (
        account-number    char(10),
        balance           integer)
```

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

    ```
    select   customer.customer-name
    from     customer
    where    customer.customer-id = '192-83-
    7465'
    ```

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

Database Users

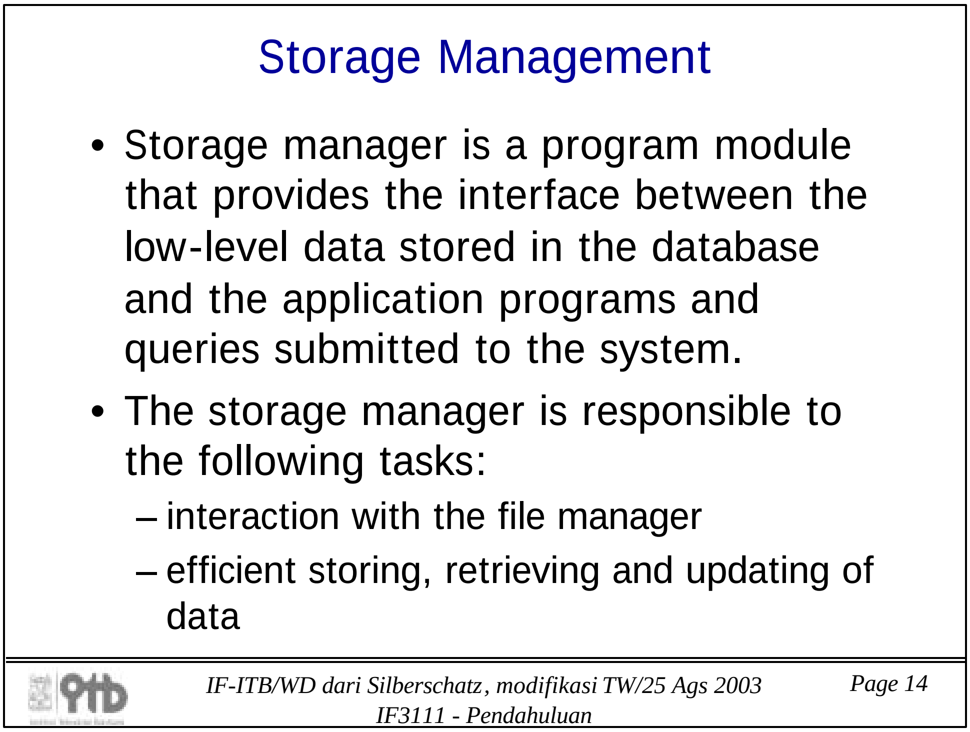
- Users are differentiated by the way they expect to interact with the system:
  - Application programmers – interact with system through DML calls
  - Sophisticated users – form requests in a database query language
  - Specialized users – write specialized database applications that do not fit into the traditional data processing framework
  - Naive users – invoke one of the permanent application programs that have been written previously

Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise’s information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure
  - Schema & physical organization modification
  - Granting user authority to access DB
  - Specifying integrity constraints

Transaction Management

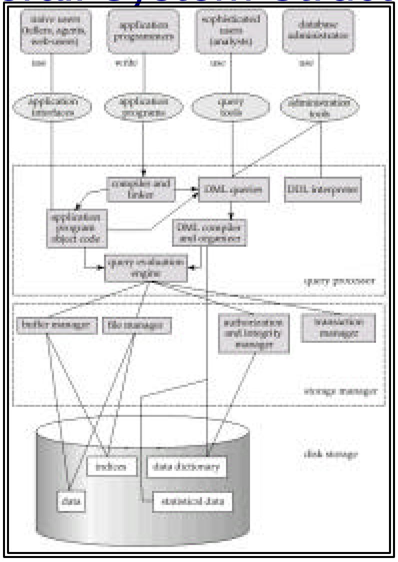
- A *transaction* is a collection of operations that performs a single logical function in a database application
- Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.



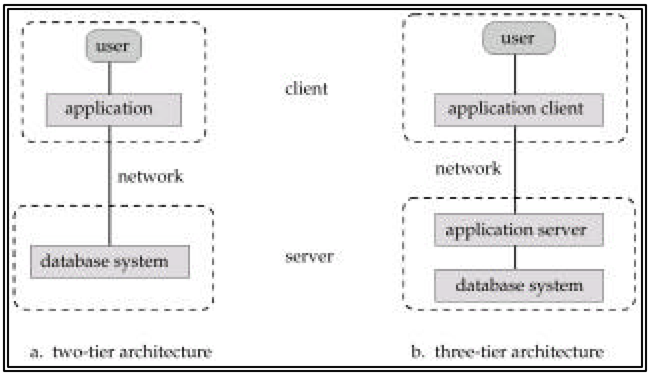
Storage Management

- Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - interaction with the file manager
  - efficient storing, retrieving and updating of data

Overall System Structure



Application Architectures



- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using “middleware”