

# Dasar-dasar Database

Umi Proboyekti

## Properti Database

Database adalah koleksi data item yang saling terkait terkelola sebagai satu unit. Beberapa definisi lain tentang database adalah :

1. Relational data structure used to store, query, and retrieve information. [fwie.fw.vt.edu/tws-gis/glossary.htm](http://fwie.fw.vt.edu/tws-gis/glossary.htm)
2. A shared collection of logically related data designed to meet the information needs of multiple users in an organization. [www.cbu.edu/~lschmitt/I351/glossary.htm](http://www.cbu.edu/~lschmitt/I351/glossary.htm)
3. an organized body of related information [wordnet.princeton.edu/perl/webwn](http://wordnet.princeton.edu/perl/webwn)
4. database is an organized collection of computer records. In libraries, one of the most common types of databases consists of records describing articles in periodicals otherwise known as a periodical index. [www.lib.utexas.edu/help/glossary/d.html](http://www.lib.utexas.edu/help/glossary/d.html)

Istilah “database” juga digunakan di dunia perpustakaan untuk menjelaskan sekumpulan informasi yang biasanya tersimpan dan dapat diakses di perpustakaan, seperti yang dijelaskan pada definisi ke 4. Untuk membuat definisi database yang kita maksud lebih jelas dan tepat, maka beberapa karakteristik dari database adalah :

1. pengelolaan melalui dengan suatu Database Management System (DBMS)
2. ada lapisan-lapisan abstraksi data
3. data fisik yang independen
4. data logis yang independen

Berikut penjelasan karakteristik-karakteristik tersebut.

## DBMS

Database Management System merupakan software yang disediakan oleh penyedia database untuk :

- mengelola dan memelihara data
- memindahkan data ke dan dari file data fisik yang dibutuhkan
- mengelola akses data yang bersamaan oleh beberapa pengguna
- mengawasi pembaharuan data dan mencegah konflik perubahan data
- mengelola transaksi sehingga perubahan data terjadi secara lengkap atau tidak terjadi perubahan jika transaksi batal atau gagal
- mendukung bahasa query
- pengawasan back up database dan pemulihan dari kesalahan
- mekanisme keamanan

## Lapisan-lapisan abstraksi data /Layers of Data Abstraction

Database memiliki kapabilitas unik unik untuk menyajikan informasi ke banyak pengguna sesuai dengan permintaan masing-masing. Pengguna di sini bisa saja manusia atau aplikasi yang masuk secara sah ke database untuk menyimpan atau mendapatkan data. Aplikasi tersebut adalah program yang dirancang untuk menyelesaikan tugas-tugas bisnis tertentu seperti sistem pemesanan barang, sistem penggajian, sistem akuntansi.

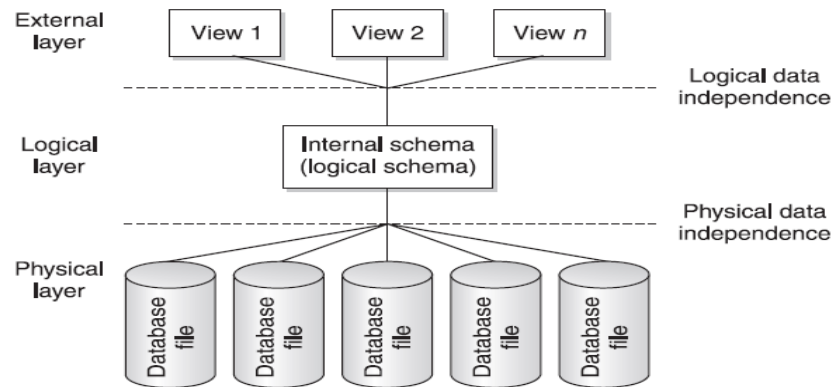
Database memungkinkan setiap pengguna (manusia atau aplikasi) mendapatkan informasi hasil pengolahan data yang berbeda satu sama lain. Yang satu minta laporan pendapatan bulan tertentu sementara yang lain mencetak transaksi hari ini, pengguna lain melakukan pencarian terhadap data tertentu. Masing-masing menggunakan data dari database yang sama. Setiap perubahan data yang dilakukan oleh salah satu, akan juga ditemukan oleh yang lain.

Lapisan-lapisan abstraksi data dapat digambarkan seperti pada Figure 1-1 dimana ada 3 lapisan yaitu lapisan fisik, lapisan logika dan lapisan luar.

a. lapisan fisik

lapisan fisik berisi file data yang menyimpan semua data untuk database. Hampis esmua DBMS yang moderen mengijinkan data base disimpan dalam multiple data files (banyak file untuk 1 database). Kecuali MS Access menyimpan database dalam 1 file sehingga tidak mampu mengatasi ukuran data yang besar, tapi cocok untuk data personal.

DBMS dan sistem operasi secara otomatis mengelola data files : membuka, menutup, membaca dan menulis data ke dan dari file. Pengguna database tidak perlu langsung berhubungan dengan lapisan fisik yaitu data files saat menggunakan database. Data-data yang tersimpan dalam file diwakili oleh lapisan berikutnya yaitu lapisan logika.



**Figure 1-1** Database layers of abstraction

b. lapisan logika

Ini adalah lapisan abstraksi pertama dalam database. Lapisan fisik bukan lapisan abstraksi karena datanya nyata, bukan abstrak. Lapisan logika terdiri dari struktur data abstrak yang sama dengan lapisan data. DBMS mengubah data di data file menjadi struktur/susunan yang sesuai. Lapisan ini sering disebut skema, istilah yang digunakan untuk koleksi semua data yang tersimpan dalam database. Pada lapisan inilah struktur tabel database yang disusun oleh manusia dan kemudian data dalam tabel-tabel itulah yang kemudian diakses oleh manusia menggunakan bahasa query yang ada di lapisan luar.

c. lapisan luar

Lapisan luar atau mmodel luar adalah layar kedua abstrak dalam database. Lapisan ini terdiri dari tampilan pengguna yang disebut subskema. Lapisan ini adalah tempat dimana pengguna dan aplikasi yang mengakses database berhubungan dan menghasilkan query ke database.

**Data fisik yang independen/ Physical Data Independence**

Kemampuan mengubah struktur file database secara fisik tanpa mengganggu pengguna dan proses yang berlaku disebut physical data independence. Seberapa besar perubahan yang dapat dilakukan tanpa berpengaruh pada lapisan logika disebut derajat physical data independence.

**Data logis yang independen/ Logical Data Independence**

kemampuan untuk membuat perubahan pada lapisan logika tanpa mengganggu penggunaan dan pross yang sedang berlaku disebut logical data independence. Perubahan pada lapisan logika juga berlaku pada lapisan fisik. Misalnya penambahan data di lapisan logika, berarti menambah data pada lapisan

fisik. Perubahan format data pada lapisan logika akan berdampak pada lapisan fisik juga. Perubahan yang dimaksud termasuk penghapusan data.

## Model Database yang Umum

### a. Flat Files

Flat files adalah file sistem operasi yang record dalam filenya tidak berisi informasi tentang struktur file atau hubungan antar record yang dikomunikasikan ke aplikasi yang menggunakannya. Flat file bukan database karena tidak masuk dalam kriteria di atas yang sudah dibahas.

Figure 1-2 adalah contoh flat file yang diambil dari MS access database milik perusahaan Northwind. Kolom judul sengaja ditambahkan untuk menjelaskan isi dari data, jadi sebenarnya kolom judul itu tidak ada. Yang tersimpan dalam file adalah record saja. Data customer disimpan pada file Customer dengan setiap record mewakili satu customer Northwind. Setiap karyawan Northwind memiliki satu record di file Employee.

#### Customer File

Customer ID	Company Name	Contact Name	Address	City	Country	Phone
ALFKI	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	Germany	030-007431
AROUT	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	UK	(171) 555-7788

#### Employee File

Employee ID	Last Name	First Name	Title
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
6	Suyama	Michael	Sales Representative

#### Product File

Product ID	Product Name	Category	Quantity Per Unit	Unit Price	Units In Stock
28	Rossle Sauerkraut	Produce	25 825 g cans	\$45.60	26
39	Chartreuse verte	Beverages	750 cc per bottle	\$18.00	69
41	Jack's New England Clam Chowder	Seafood	12 12 oz cans	\$9.65	85
46	Spegesild	Seafood	4 450 g glasses	\$12.00	95
52	Filo Mix	Grains/Cereals	16 2 kg boxes	\$7.00	38
63	Vegie-spread	Condiments	15 625 g jars	\$43.90	24

#### Order File

Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date	Ship Via	Freight
10643	ALFKI	6	25-Aug-1997	22-Sep-1997	02-Sep-1997	Speedy Express	\$29.46
10692	ALFKI	4	30-Oct-1997	03-Oct-1997	31-Oct-1997	United Package	\$61.02
10793	AROUT	3	24-Dec-1997	21-Jan-1998	08-Jan-1998	Federal Shipping	\$4.52

#### Order Detail File

Order ID	Product ID	Unit Price	Quantity	Discount
10643	28	\$46.50	15	25%
10643	39	\$18.00	21	25%
10643	46	\$12.00	2	25%
10692	63	\$43.90	20	0%
10793	41	\$9.65	14	0%
10793	52	\$7.00	8	0%

**Figure 1-2** Flat file order system

Data pemesanan (order) disimpan dalam 2 flat file: file Order berisi record pesanan customer yang terdiri dari ID customer, nama karyawan yang menerima order, tanggal pesanan dsb. Sementara di file Order Detil berisi item tiap pesanan. Satu pesanan dapat terdiri dari beberapa barang dan detil tiap barang seperti harga ID produk, jumlah barang yang dipesan dan diskon.

Northwind memiliki aplikasi yang mampu menghasilkan informasi berdasarkan data dari flat file nya. Aplikasi ini harus menghubungkan data di antara 5 file dengan membaca 1 pesanan/order dan mengikuti langkah ini:

1. gunakan customer ID untuk mencari nama dari customer di file Customer
2. gunakan employee ID untuk mencari nama karyawan yang bersangkutan di file Employee
3. gunakan order ID untuk mencari item barang yang sesuai di file Order Detail
4. dan untuk setiap item gunakan product ID untuk mencari nama produk yang sesuai di file Produk.

Cara ini memang rumit untuk membuat satu laporan pemesanan dari data di flat file. Alternatif lain adalah menggabungkan semua data menjadi satu file data. Alternatif ini akan menyederhanakan pencarian data, tapi penambahan data harus lengkap, artinya kita tidak dapat menambahkan data customer jika customer itu tidak memesan barang.

#### b. Hierarchical Model

Model ini menyusun record-record dalam suatu hirarki seperti struktur organisasi. Setiap file dari flat file menjadi tipe record atau node dalam hirarki, dan untuk memudahkan disebut record. Record-record terhubung melalui pointer yang berisi alamat record yang terkait. Pointer memberi tahu sistemkomputer dimana record tersebut tersimpan secara fisik, seperti alamat yang mengacu ke suatu gedung tertentu di dunia nyata. Setiap pointer membuat hubungan parent-child, yang juga disebut one-to-many relationship. Satu parent bisa punya banyak child, tapi setiap child memiliki 1 parent. Misal 1 manager memimpin banyak karyawan, tapi 1 karyawan memiliki 1 manager.

Model ini menyebabkan masalah dalam kasus Northwind. Satu order tidak hanya punya satu customer, tapi juga punya satu karyawan yang menerima order tersebut. Pada Figure 1-3, Order dan Employee serta Order Detail dan Product tidak terhubung dengan garis tegas karena tidak ada pointer yang menghubungkan mereka, sekalipun seharusnya berhubungan. Ini akibat aturan yang tegas : 1 child punya 1 parent.

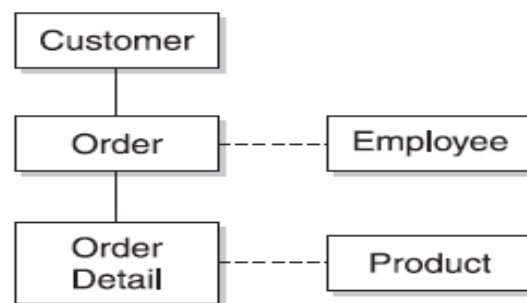
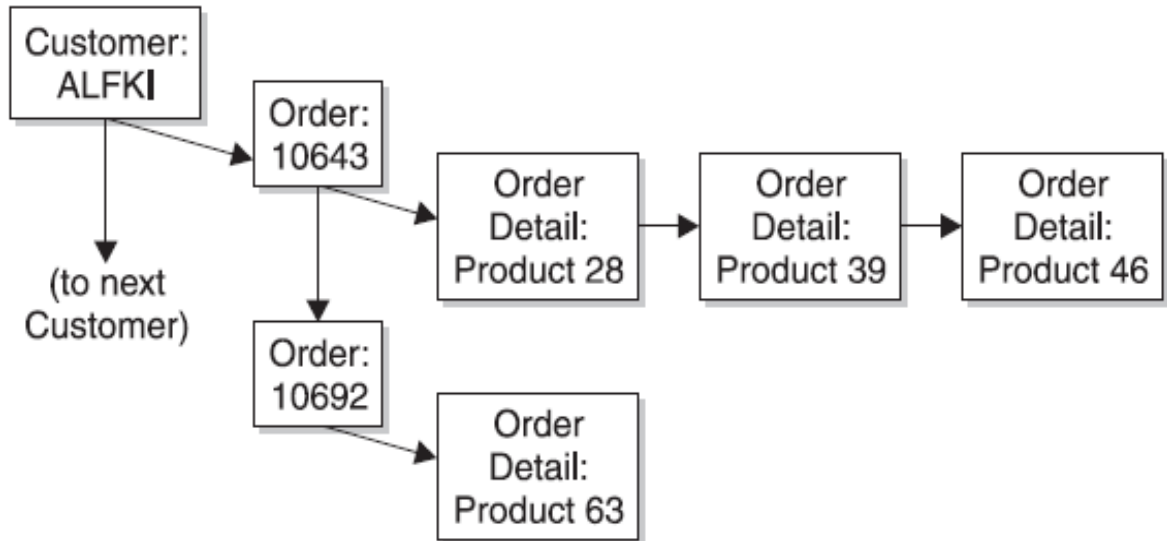


Figure 1-3. Hierarchical Model Structure for northwind

Figure 1-4 menunjukkan isi dari record yang terpilih dalam model hirarki Nortwind. Record customer ALFKI menunjuk ke order pertama (ID 10643) dan order tersebut memiliki pointer menunjuk ke order berikutnya (ID 10692). Masing-masing order menunjuk ke detil ordernya. Karena tidak ada hubungan dengan file Employee dan Product maka tidak dapat mendapatkan informasi yang berkaitan dengan record pemesanan di 2 file tersebut.



Figures 1-4. Hierarchical model record contents for northwind

### c. Relational Model

Model ini berusaha mengatasi masalah pada model sebelumnya yang tidak fleksibel. Model relational database mampu menghubungkan record-record yang dibutuhkan. Model ini dibangun sedemikian rupa sehingga permintaan atau query dapat bekerja dengan sekumpulan data, tidak dengan satu record seperti pada model hierarchial.

Model relational menyajikan data dalam tabel dua dimensi, seperti spreadsheet hanya saja tidak dipresentasikan dalam tabel seperti spreadsheet tetapi menggunakan kombinasi atau joining relational tabel untuk menghasilkan view atau hasil tampilan.

Figure 1-5 memberikan gambaran bagaimana hubungan antar tabel yang diwakili dengan garis antar tabel. Hubungan yang dapat terjadi adalah : one-to-many relationship. Garis tunggal menjelaskan ONE dan garis bercabang menjelaskan MANY. Contoh ONE customer berelasi dengan MANY order dan ONE order berelasi dengan MANY order details. Diagram yang dipakai pada figure tersebut adalah ERD atau Entity-Relationship Diagram.

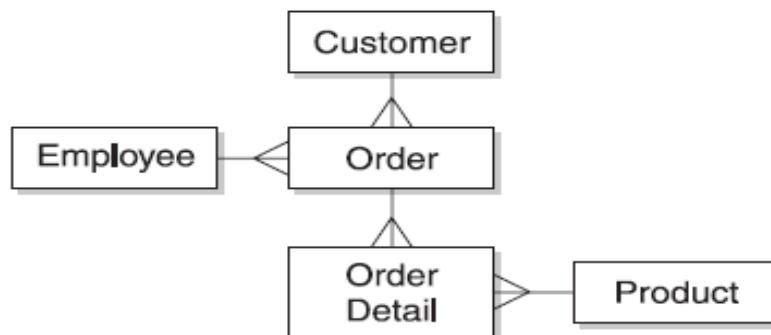


Figure 1-5. Relational model structure for Northwind

Figure 1-6 memberikan gambaran bagaimana relasi antara 3 tabel: Customer, Order dan Employee saling terhubung sehingga memungkinkan menghasilkan informasi pesanan barang tiap customer lengkap dengan data karyawan yang menerima pesanan.

### Customer Table

Customer ID	Company Name	Contact Name	Address	City	Country	Phone
ALFKI	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	Germany	030-007431
AROUT	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	UK	(171) 555-7788

### Order Table

Order ID	Customer ID	Employee ID	Order Date	Required Date	Shipped Date	Ship Via	Freight
10643	ALFKI	6	25-Aug-1997	22-Sep-1997	02-Sep-1997	Speedy Express	\$29.46
10692	ALFKI	4	30-Oct-1997	03-Oct-1997	31-Oct-1997	United Package	\$61.02
10793	AROUT	3	24-Dec-1997	21-Jan-1998	08-Jan-1998	Federal Shipping	\$4.52

### Employee Table

Employee ID	Last Name	First Name	Title
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
6	Suyama	Michael	Sales Representative

Figure 1-6 Relational table contents for Northwind

#### d. Object-Oriented Model

Model ini berusaha menjawab kesulitan RDBMS (Relational Database Management Systems) untuk mengatasi tipe data yang kompleks seperti file citra, file gambar dan file audio-video. Ini semua karena adanya Internet dan WWW yang memungkinkan pengiriman data yang kompleks itu.

Object adalah pengelompokan logis dari data yang berelasi dan program logis yang mewakili hal nyata di dunia seperti customer, employee, order, product.

Item data seperti : customer ID, customer name disebut variabel di model OO dan disimpan dalam tiap objek. Variabel-variabel hanya bisa diakses melalui method. Sementara istilah method adalah aplikasi program logis yang menggunakan object tertentu dan menyediakan fungsi tertentu seperti check customer credit card limit, update alamat customer.

Figure 1-7 menunjukkan objek Customer sebagai contoh implementasi OO. Pada inti lingkaran terdapat variabel-variabel tentang objek dan kumpulan variabel itu dikelilingi oleh method yang menunjukkan bahwa hanya dengan melalui method saja variabel tersebut dapat diakses. Ini adalah konsep encapsulate, salah satu konsep penting di OO model.

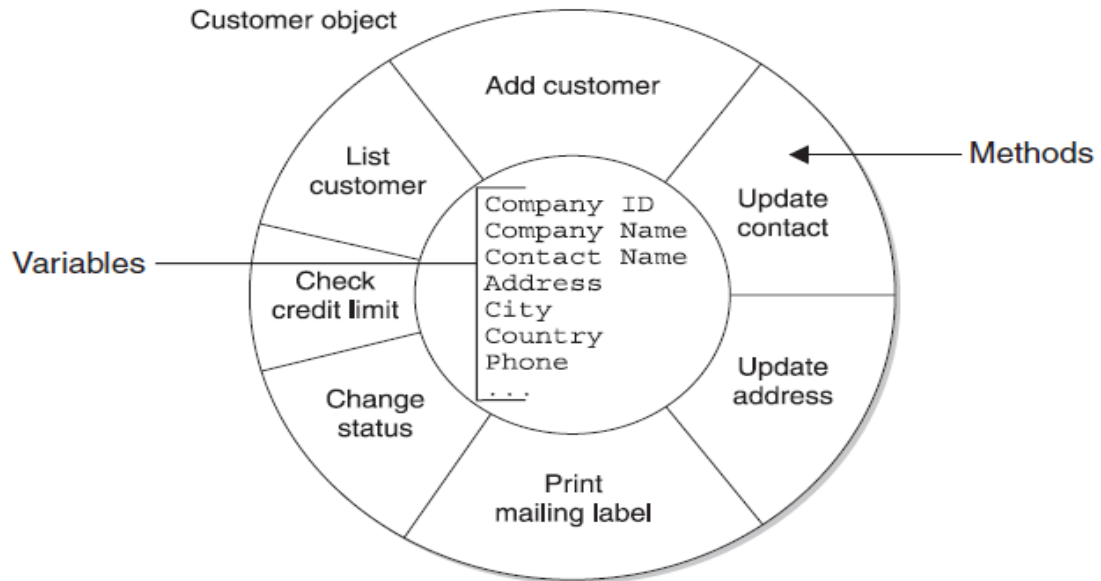


Figure 1-7 The Anatomy of an object

Diadaptasi dari :  
 Oppel, Andy. **Databases Demystified: A self-teaching guide**. McGraw-Hill. Ca. 2004.