

# 12

## Database dan Sistem Manajemen Database

Setelah membaca bab ini anda diharapkan dapat memahami hal-hal sebagai berikut:

- Database*
- Organisasi data pada database tradisional
- Organisasi data pada database modern
- Sistem *database*
- Abstraksi *data*
- Model-model organisasi *data*

## Pendahuluan

Didalam suatu sistem informasi berbasis komputer *data* yang masuk ada yang langsung diolah ada pula yang disimpan terlebih dahulu untuk diolah saat diperlukan. *Data-data* tersimpan di media penyimpanan tambahan (*secondary storage*) relatif permanen. Permasalahan yang muncul adalah bagaimana *data-data* tersebut disimpan dalam media penyimpanan dan bagaimana *data-data* tersebut berhubungan satu sama lain. Bagian buku ini akan menjelaskan tentang organisasi *data* tradisional dan modern serta berbagai macam bentuk organisasi *data* yang banyak digunakan saat ini.

## 12.1 Database

---

*Data* adalah fakta baik dalam bentuk angka-angka, huruf-huruf atau apapun yang dapat digunakan sebagai *input* dalam proses untuk menghasilkan informasi. *Data* terdiri dari tiga macam, ketiga macam *data* tersebut adalah:

1. '**Input data**' adalah *data* yang dimasukkan ke dalam sistem informasi.
2. '**Output data**' merupakan keluaran dari sistem informasi
3. '**Database**' merupakan kumpulan *data-data* yang tersimpan didalam media penyimpanan di suatu perusahaan (arti luas) atau di dalam komputer (arti sempit)

## 12.2 Organisasi Data Pada Database Tradisional

---

**Organisasi data** pada *database* tradisional memiliki tujuan agar sistem Informasi secara efektif memberikan informasi yang akurat, relevan, tepat waktu dan lengkap

Organisasi *data* pada *data base* tradisional memiliki tujuan agar sistem Informasi secara efektif memberikan, kepada para pemakai sistem informasi, informasi yang akurat, relevan tepat waktu dan lengkap. Informasi ini merupakan hasil pengolahan *data* yang disimpan dalam *file-file* komputer.

### 12.2.1. Hirarki Data

Dalam konsep *database*, *data* memiliki tingkatan yang dikenal sebagai hirarki *data* yang terdiri dari *Bit, Byte, Filed, Record, File, Data base*

- ❑ **Bit** adalah unit terkecil dari *data*. Bit ini menggambarkan sinyal 0 dan satu, dimana 0 berarti tidak ada arus listrik dan satu ada arus listrik.
- ❑ **Byte** adalah kumpulan dari *bit-bit* yang membentuk sebuah karakter
- ❑ **Field/Elemen data (data element)** adalah kumpulan karakter-karakter yang membentuk suatu kata atau sekelompok kata/angka.
- ❑ **Record** merupakan kumpulan dari *field-field* yang secara logika berhubungan.
- ❑ **File** adalah kumpulan dari *record-record* yang berhubungan dengan suatu subjek tertentu.
- ❑ **Data base** adalah kumpulan dari *data-data* yang tersimpan dalam *file-file*

Table 12.1. Hirarki data dalam konsep database tradisional

Hirarki data		Contoh		
Database		File Gaji	File Alamat	File Tarip
File	File Alamat			
	Nama	Alamat		umur
	Dine	Jl. Panghegar 18 Bandung		28
	Shelly	Jl. Polisi 8 Bandung		30
Maman	Jl. Thamrin 12 Jakarta		20	
Record	Nama	Alamat		umur
	Doni	Jl. Dipatiukur No 350 Bandung		18
Field	Bambang Sutopo (Nama pada sebuah field nama)			
Byte	0100 0001 (Huruf A dalam ASCII)			
Bit	0			

**Menentukan Alamat Record**

Umumnya dikenal tiga cara dalam menentukan alamat record pada DASD, cara tersebut adalah:

1. *Direct /Random* (Secara Langsung).
2. *Hashing* (Pakai rumus).
3. *Indexed Sequential* (Urutan berdasarkan index).

Ada tiga cara menentukan alamat record, yaitu:  
 -*Direct /Random*  
 -*Hashing*  
 -*Indexed sequensi*

**Penentuan alamat cara langsung (Direct/Random)**

Dengan cara ini, alamat record pada DASD menggunakan nomor kunci record (*record key*). Kunci record itu sendiri merupakan suatu field/elemen data/attribute/identifier yang mewakili record suatu file.

Hirarki data: Contoh

Database

Tabel 12.2 Contoh key record dan alamat record

Nama File master : Siswa

NPM	Nama	Alamat
B1A98001	Bambang sutopo	Jl. Ir. H. Juanda 111
B1A98002	Fuad lufti	Jl. Reformasi 80
B1A98003	Crisanti Nova	Jl. Jati 12

File

File Alamat  
 Nama  
 Dine  
 Shelly  
 Maman

Pada file master siswa diatas yang menjadi kunci record adalah field NPM, sedangkan alamat record menggunakan nilai dari tiga angka dibelakang field NPM yaitu 001, 002, 003. Berikut ini adalah contoh lain dari key record:

Record

Nama  
 Doni

Field

Bambang :

Byte

0100 0001

Bit

0

Tabel 12.3 Contoh key record sama dengan alamat record

Nama file : Transaksi

No_transaksi	Tanggal_transaksi	Jenis_transaksi	Jumlah
000001	21-12-1998	Tunai	8000
000002	21-12-1998	Kredit	15000

Pada file transaksi ini kunci record adalah No\_transaksi sama dengan alamat record. Baik kunci record yang dicontohkan pada tabel 12.2 maupun tabel 12.3 keduanya digunakan untuk menunjukkan alamat record dalam DASD, Hanya alamat record pada tabel 12.2 harus diproses dahulu untuk menjadi alamat record, tapi bentuk pada tabel 12.2 adalah bentuk yang banyak dipakai.

### Menentukan Alamat Record Dengan Hashing

**Hashing** merupakan teknik yang digunakan untuk menentukan alamat record dengan menggunakan rumus

**Hashing** memberikan banyak kesulitan dalam menambah dan menghapus data

Menentukan alamat record dengan menggunakan hashing muncul karena menentukan alamat dengan sequensial memberikan akses yang lama sedangkan dengan menggunakan cara akses langsung sulit menentukan nomor record tertentu yang berhubungan. Misalnya seorang pelanggan bernama Doni sering membeli barang ke perusahaan secara kredit. Di perusahaan terdapat ribuan pelanggan dengan tingkat frekuensi pembelian yang berbeda. Masalahnya sekarang pada record nomor berapa saja Doni ini pernah melakukan transaksi sehingga saat di akses data-nya bisa muncul dengan cepat, artinya komputer tidak membaca record-record yang tidak diperlukan. Maka untuk ini bisa digunakan teknik hashing (menggunakan rumus) agar bisa diketahui lokasi penyimpanan record-record yang berhubungan dengan Doni.

Akan tetapi menentukan alamat dengan menggunakan metode ini memberikan banyak kesulitan terutama yang berhubungan penambahan data baru (Insert) atau penghapusan data yang ada (Delete).

Tabel 12.4 Pengalamatan dengan Rumus (Hashing)

Kode	Nama	R_A	No	Kode1	Kd_Brg	Nama	Jl	R_N
98001	Dani	1	1	98001	235001	Sepatu	1	2
98002	Charles	3	2	98001	233034	Kemeja	2	6
			3	98002	350081	Sandal	1	4
			4	98002	450002	Kompom	1	5
			5	98002	350001	Sabun	3	
			6	98001	350004	Odol	2	

R\_A = Alamat record awal/nomor record awal

R\_N = Alamat record selanjutnya/nomor record selanjutnya

## Menentukan Alamat Record Dengan Menggunakan Urutan Yang Diindek (Indexed Sequential).

*File* dapat disusun urutannya dengan menggunakan *index*. Melalui metode ini *record-record* dicatat pada piringan secara berurutan. *File-file index* yang berisi kunci-kunci *record* disimpan dalam *file* yang terpisah. *File index* ini selalu mengingat alamat yang terakhir. Dan untuk memeliharanya agar *file index* selalu sesuai dengan susunan yang ada di *file master/transaksi* maka kita harus selalu melakukan *reindex* setiap perubahan alamat terjadi.

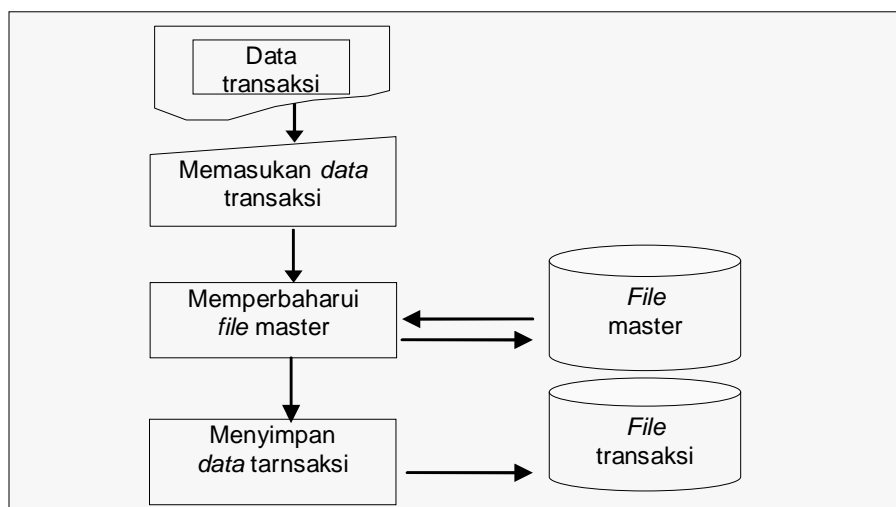
Dari penjelasan diatas, pada era *data base* tradisional, *Programmer* yang dibantu oleh analis sistem sangat disibukan oleh kegiatan dalam menentukan alamat *data* yang paling optimal. Sekarang permasalahan itu sudah sangat berkurang bahkan dihilangkan sehingga para spesialis sistem informasi bisa lebih memfokuskan pikiran kepada masalah-masalah bisnis.

**Indexed Sequential** adalah menentukan alamat *record* dengan menggunakan index

## Memperbaharui File Master Pada DASD

1. Langkah 1 - *Data* transaksi dimasukkan melalui suatu terminal dimanapun terminal itu berada, asalkan *online*, dan setiap *data* transaksi yang masuk sudah termasuk didalamnya alamat *record* yang ada di *file master*.
  - 1.1. Alamat *record* akan memanggil *file record* yang ada di *file master*
  - 1.2. *Data* transaksi akan memperbaharui *data* yang ada di *file master* berdasarkan *data* baru yang masuk melalui *file* transaksi
2. Langkah 2, *record* yang telah diperbarui ditulis ulang ke lokasi asalnya. Pada saat yang bersamaan *data* transaksi disimpan.

Gambar 12.1. Memperbaharui *data* pada DASD



## 12.2.2. Sistem Pengolahan

Ada dua cara mengolah *data* yang biasa dilakukan dalam sistem manajemen *data* saat ini, yaitu pengolahan secara *Batch* dan pengolahan secara *On-line*.

**Pengolahan secara *batch*** merupakan pengolahan *data* dengan menumpuk dahulu *data* yang mau diolah dan kemudian diolah pada saat tertentu

### Pengolahan Secara Batch

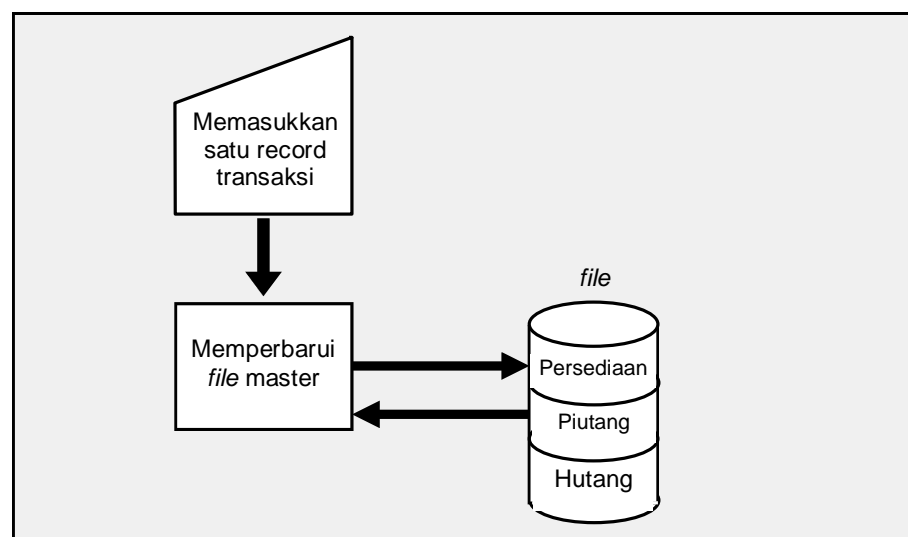
Pengolahan secara *batch* (Penumpukan) merupakan sistem pengolahan *data* transaksi dengan cara mengumpulkan terlebih dahulu *data* transaksi yang terjadi, kemudian pada waktu yang telah ditentukan secara sekaligus memproses *data* transaksi tersebut, biasanya sambil memperbaharui *file* master.

### Pengolahan Secara On-line

**Pengolahan secara *on-line*** merupakan pengolahan secara langsung begitu *data* dimasukkan kedalam suatu sistem informasi.

Pengolahan Secara *on-line* (Pengolahan langsung) merupakan sistem pengolahan *data* transaksi dimana setiap *data* yang masuk secara langsung satu persatu diolah. Pada saat yang bersamaan biasanya juga dilakukan proses untuk memperbaharui *file* master. Istilah lain yang digunakan adalah pemrosesan transaksi. Pengolahan *on-line* dikembangkan untuk memperoleh informasi yang selalu mutakhir.

Gambar 12.2 Pengolahan secara on line



### Masalah Pada Organisasi Data Tradisional

Banyak organisasi mengawali pengolahan *data* dalam skala kecil dengan mengotomatisasikan satu aplikasi tertentu pada bagian tertentu. Pada bagian lain, sistem organisasi tersebut cenderung tumbuh tersendiri tidak tergantung kepada rencana perusahaan secara keseluruhan. Akibatnya, bagi perusahaan yang memiliki bagian/departemen yang banyak, banyak bagian-bagian dari organisasi tersebut yang membuat sistemnya sendiri-sendiri yang terisolasi. Misalnya bagian tabungan di salah satu bank menyim-

pan *data* langganan dan rekening tabungan didalam *file* yang permanen. Sistem informasi pada bagian tersebut memiliki sejumlah program aplikasi, seperti program untuk mendebet dan mengkredit sebuah rekening, menambah rekening baru, mencari saldo rekening tertentu dan membuat laporan bulanan.

Semua program aplikasi ini dibuat oleh *programer* aplikasi sesuai dengan permintaan organisasi perbankan. Seandainya muncul peraturan pemerintah yang baru yang mengharuskan bank tersebut untuk menawarkan rekening koran maka akibatnya perusahaan harus membuat *file master* baru yang berisi informasi tentang semua rekening koran yang dikelola oleh bank, dan program aplikasi yang baru.

Akibat ini semua, akhirnya akan terdapat banyak sekali *file-file data* dan program yang ditambahkan kedalam sistem. Proses ini menimbulkan beberapa masalah yang antara lain:

**Masalah pada *data base* tradisional** adalah :

- *Data* rangkap dan tidak konsisten
- Kesulitan dalam akses *data*
- *Data* terisolasi sulit di akses bersamaan
- Masalah keamanan
- Masalah integritas

- ❑ ***Data* rangkap dan tidak konsisten (*Redundancy* dan *inconsistency*)** - terjadi karena *file-file* dan program dibuat oleh *programer* yang berbeda selama beberapa waktu, *file* seringkali memiliki format yang berbeda dan program mungkin dibuat dalam bahasa pemrograman yang berbeda. Karena itu beberapa bagian dari informasi terduplikasi pada beberapa *file*. Seperti alamat dan nomor telpon konsumen tertentu mungkin tertulis di *file* konsumen dan *file* piutang. *Data* berlebihan yang disebabkan oleh adanya duplikasi *data* menyebabkan selain volume penyimpanan lebih besar dan biaya lebih tinggi juga menyebabkan akses *data* menjadi lebih lambat. Bila *data* tertentu memerlukan perubahan akan sangat sulit dilakukan karena *data* tersebar diberbagai *file*. Akibat dari itu semua kalau ada perubahan yang tidak seragam misalnya alamat berubah maka *data* yang tersimpan menjadi tidak konsisten. Berikut ini contoh dari perangkapan *data*.

Tabel 12.5 File konsumen

Kode	Nama	Alamat	Telpon
K98001	Agus saptono	Jl. Kebumen 80	250480
K98002	Hasan sadikin	Jl. Supratman 15	2506150

Tabel 12.6 File piutang

Kode	Nama	Alamat	Telpon	No_Bukti
K98001	Agus saptono	Jl. Kebumen	250480	J_000885
K98002	Hasan sadikin	Jl. Dipatiukur 85	250820	J_000886

Contoh diatas menunjukkan dua *file* konsumen dan piutang. Dalam *file* piutang tersimpan kembali *data-data* konsumen. Sehingga ada dua *data* konsumen (rangkap) akibatnya bila terjadi pemberitahuan dari konsumen bahwa alamatnya

sudah pindah, dan perubahan itu hanya dilakukan terhadap *file* piutang maka *data* tentang konsumen jadi tidak konsisten antara *file* yang satu dengan *file* lainnya. Salah satu prinsip dalam sistem database tidak boleh ada duplikasi *data*.

- ❑ **Kesulitan dalam mengakses data** - misalnya seorang karyawan perusahaan ingin mencari *data* konsumen yang berlokasi di jalan Soekarno-Hatta. Bentuk Permintaan ini tidak terantisipasi sebelumnya, karena itu tidak ada program aplikasi yang dapat memenuhi permintaan ini, diluar ini sebenarnya ada program aplikasi yang dapat menghasilkan daftar seluruh konsumen. Karyawan perusahaan tersebut sekarang punya dua pilihan. Pertama mencetak semua daftar pelanggan dan memilihnya secara manual satu persatu atau membuat program aplikasi baru untuk kebutuhan tersebut. Kedua alternatif ini sudah barang tentu tidak disukai. Kesimpulan dari bagian ini adalah sistem pencarian *data* harus dibuat untuk penggunaan yang umum dilakukan.

Tabel 12.7. Daftar konsumen

PT. ABC  
Jl. Reformasi 22  
Bandung

#### Daftar Konsumen

Kode	Nama	Alamat	Telpon
K98001	Agus saptono	Jl. Kebumen 80	250480
K98002	Hasan sadikin	Jl. Supratman 15	2506150
K98003	Fuad lufti	Jl. Pungkur 28	2508812
K98004	Kursin senjaya	Jl. Ir. H. Juanda 200	2508820
K98005	Hasan basri	Jl. Supratman 85	2506150

Tabel diatas menggambarkan sebagian daftar karyawan yang jumlahnya sekitar 2000 orang. Untuk mencari seseorang yang beralamat di jalan Sukarno-Hata akan sulit karena karyawan perusahaan tersebut harus mencari satu persatu dari daftar diatas, dan untuk mencetaknya perlu biaya yang cukup mahal. Seperti dijelaskan diatas agar supaya keadaan ini tidak terulang lagi kiranya perlu dibuat program baru untuk melakukan pencarian secara spesifik tersebut. Masalah lainnya adalah siapa yang bisa menentukan bahwa pada suatu saat nanti akan diperlukan informasi tertentu?. Inilah masalah utama lainnya yang sering dihadapi dalam penyusunan *database*. Menentukan informasi apa yang diperlukan sehingga bisa disusun struktur datanya yang bisa memenuhi kebutuhan informasi tersebut.

- ❑ **Data terisolasi** - maksudnya *data* tersebar pada *file-file* di berbagai bagian dan apabila *file-file* tersebut memiliki format yang berbeda, ini akan sangat menyulitkan saat membuat



program aplikasi baru untuk membaca *data* yang diperlukan, karena ragu *data* mana yang harus dipakai dan yang dianggap benar.

Tabel 12.8 File konsumen

Kode	Nama	Alamat	Telpon
K98001	Agus saptono	Jl. Kebumen 80	250480
K98002	Hasan sadikin	Jl. Supratman 15	2506150

Tabel 12.9. File piutang

Kode	Nama	Alamat	Telpon	No_Bukti
K98001	Agus saptono	Jl. Kebumen	250480	J_000885
K98002	Hasan sadikin	Jl. Dipatiukur 85	250820	J_000886

Dua tabel diatas hanya menggambarkan satu contoh saja yang menyebabkan *data* terisolasi. Tabel-tabel tersebut menggambarkan *data* konsumen yang ada dalam dua *file* tapi formatnya yang berbeda. Akibatnya, *data* hanya bisa dipakai untuk kebutuhan tertentu pada aplikasi tertentu. *Data* tidak bisa dipakai untuk berbagai kebutuhan aplikasi yang berhubungan karena tidak ada kepastian format mana yang akan dipakai.

- ❑ **Data Sulit Diakses secara bersamaan** - untuk memperbaiki semua kinerja dan respon yang cepat dari suatu sistem dilakukan dengan memberikan kemungkinan beberapa pemakai untuk meng*update data* secara bersamaan. Akibatnya, perubahan *data* didalam database begitu cepat dan sulit untuk diawasi. Table 12.8 dan 12.9 menunjukkan kode K98002 memiliki alamat yang berbeda akibat akses dilakukan secara bersamaan terhadap beberapa *file* yang fungsinya sama. Seharusnya hanya ada satu *file*
- ❑ **Masalah keamanan data** - karena tidak semua pemakai (*user*) *data base* dapat mengakses semua *data*. Sebagai contoh bagian gaji di suatu perusahaan hanya dapat melihat *file* yang berhubungan dengan karyawan perusahaan. Mereka tidak berhak untuk mengakses *file* yang berhubungan dengan penjualan. Bila program aplikasi baru ditambahkan kedalam sistem informasi perusahaan maka akan sulit menerapkan sistem keamanannya kalau semua *data* terisolasi.
- ❑ **Masalah integritas** - nilai *data* yang disimpan didalam *data base* perusahaan harus sesuai dengan cara batasan-batasan tertentu yang telah ditetapkan. Sebagai contoh rekening bank tidak boleh kurang dari Rp.100.000. Batasan ini diterapkan di dalam sistem dengan menambahkan kode tertentu kedalam

berbagai aplikasi program terkait. Akan tetapi ketika batasan baru akan ditambahkan, hal ini sangat sulit untuk dilakukan karena untuk menerapkan batasan yang baru tadi harus dilakukan perubahan-perubahan dalam program terkait tadi secara keseluruhan. Permasalahan menjadi semakin menumpuk pada saat batasan tersebut akan mempengaruhi beberapa item *data* lama yang tersebar diberbagai *file*.

**Sistem *database* modern** memberikan banyak keuntungan bagi sistem informasi manajemen

Masalah masalah diatas sangat dominan pada sistem *data - base* tradisional. Dengan sistem database yang lebih maju (modern), sistem tersebut akan memberikan banyak keuntungan bagi sistem informasi manajemen yang diantaranya adalah:

- ❑ **Data yang berlebihan dapat dikurangi** sehingga biaya penyimpanan dan waktu bisa diperkecil.
- ❑ **Ketidak konsistenan *data* dapat dihindari** sehingga *data* lebih mudah di akses dan informasi akan lebih akurat.
- ❑ **Data dapat dipakai bersama** sehingga setiap bagian akan memperoleh informasi yang sama .
- ❑ **Standarisasi dapat dilakukan** sehingga memudahkan dalam membaca dan memasukan *data*.
- ❑ **Keamanan dapat diterapkan** sehingga tingkat akurasi informasi manajemen yang dihasilkan menjadi lebih tinggi
- ❑ **Keterpaduan dapat dijaga** sehingga meningkatkan integritas suatu sistem informasi.
- ❑ **Konflik interest dapat diseimbangkan** sehingga sistem informasi bisa berjalan tanpa rintangan.

### 12.3 Organisasi Data Pada Database Modern

---

Agar *data* atau informasi sampai keberbagai sasaran dari berbagai sumber maka *data-data* yang masuk keperusahaan harus dikelola dengan baik, Pengelolaan *data* atau informasi ini bisa dilihat dari arti luas dan sempit.

Manajemen *data* dalam Arti sempit, perusahaan dikatakan sudah mengelola *data/informasi* dengan baik bila sudah menggunakan atau menerapkan *DBMS (Database Management System)*.

Manajemen *data* dalam arti luas, perusahaan sudah mengelola *data/informasi* dengan baik bila sudah menggunakan atau menerapkan IRM (Information resource management/Manajemen sumber daya informasi) yang komponennya meliputi *hardware, software, brainware, prosedur* dan *database*. Dalam penjelasan

berikut ini akan di uraikan mengenai sistem manajemen database atau mengambil konsep manajemen *data* dalam arti sempit. Buku ini tidak akan menjelaskan bagaimana manajemen *data* dalam arti luas.

### Kegiatan Manajemen Data

- ❑ **Mengumpulkan *data*** - *data-data* yang diperlukan dikumpulkan dari berbagai fakta dan selanjutnya dicatat dalam suatu formulir yang disebut dokumen sumber (*sources document*). Dokumen sumber ini akan berfungsi sebagai *input* bagi sistem kalau dimasukan kedalam sistem. Contoh dari dokumen sumber ini misalnya suatu dokumen yang menjelaskan adanya pesanan penjualan.
- ❑ **Menjaga dan mengadakan pengujian terhadap Integritas *data*** - *data* yang akan diolah diperiksa untuk meyakinkan konsistensi dan akurasi berdasarkan suatu aturan dan batasan yang telah ditentukan sebelumnya.
- ❑ **Menyimpanan *data*** - *data-data* yang berasal dari dokumen sumber dan telah dimasukan kedalam sistem komputer (Sistem Informasi) disimpan pada suatu media seperti pita magnetik atau piringan magnetik.
- ❑ **Memelihara *data*** - pemeliharaan *data* meliputi aktivitas-aktivitas penambahan *data* baru, mengubah *data* yang ada, dan menghapus *data* yang tidak lagi diperlukan agar *data* atau informasi yang ada tetap mutakhir.
- ❑ **Mengamankan *data*** - aktivitas ini merupakan upaya untuk menjaga agar *data* terhindar dari penghancuran, kerusakan, atau penyalahgunaan baik yang disengaja ataupun yang tidak disengaja.
- ❑ **Mengorganisasikan *data*** - merupakan kegiatan menyusun *data* sedemikian rupa sehingga memenuhi kebutuhan informasi pemakai.
- ❑ **Mencari *data*** - *data* yang tersimpan di dalam media penyimpanan harus dicari sedemikian rupa sehingga tersedia pada saat diperlukan.

**Manajemen *data*** meliputi :

- Mengumpulkan data
- Menjaga dan mengadakan pengujian terhadap Integritas data
- Menyimpanan *Data*.
- Memelihara Data
- Mengamankan *Data*.
- Mengorganisasikan data
- mencari data

## 12.4 Sistem Database

---

Dalam database modern mulai dipakai konsep sistem. Date menyatakan bahwa sistem database pada dasarnya merupakan sistem pencatatan dengan menggunakan komputer yang memiliki tujuan untuk memelihara informasi agar selalu siap pada saat diperlukan. Informasi yang dianggap penting dapat apa saja, bisa informasi untuk keperluan pribadi dalam kaitan dengan organisasi

atau untuk keperluan organisasi secara keseluruhan. Sistem *data base* seperti halnya sistem informasi manajemen memiliki beberapa komponen seperti: *Data, Hardware, Software* dan Pemakai

Karena sistem *database* merupakan bagian dari Sistem Informasi manajemen (SIM), maka komponen-komponen dalam sistem *database* merupakan bagian dari komponen SIM.

### 12.4.1. Komponen Data

*Data* dalam *database* bisa tersimpan dalam :

- Komputer kecil (PC) - Biasanya untuk *single user*
- Komputer mini - Untuk *multi user*
- Komputer *Mainframe* - Untuk *multi user*

Secara umum dapat dikatakan bahwa *data* dalam sistem *database* harus selalu terintegrasi (*integrated*) dan dapat diakses oleh siapa saja yang berhak (*shared*).

- Terintegrasi** - artinya *data base* dapat dianggap sebagai perpaduan secara logis dari beberapa *file data* yang berbeda.

Tabel 12.10. *File* karyawan

Kode	Nama	Alamat	Telpon
K98001	Agus saptono	Jl. Kebumen 80	250480
K98002	Hasan sadikin	Jl. Supratman 15	2506150

**Data dalam sistem *database*** harus selalu terintegrasi dan dapat diakses oleh siapa saja yang berhak

Tabel 12.11. *File* pendaftaran

Kode	Pendidikan	Nama Orang tua	Pekerjaan orang tua
K98001	S1/Elektro	Rd. mas haryo	Wiraswasta
K98002	S1/Pertanian	Saleh sumarjan	Wiraswasta

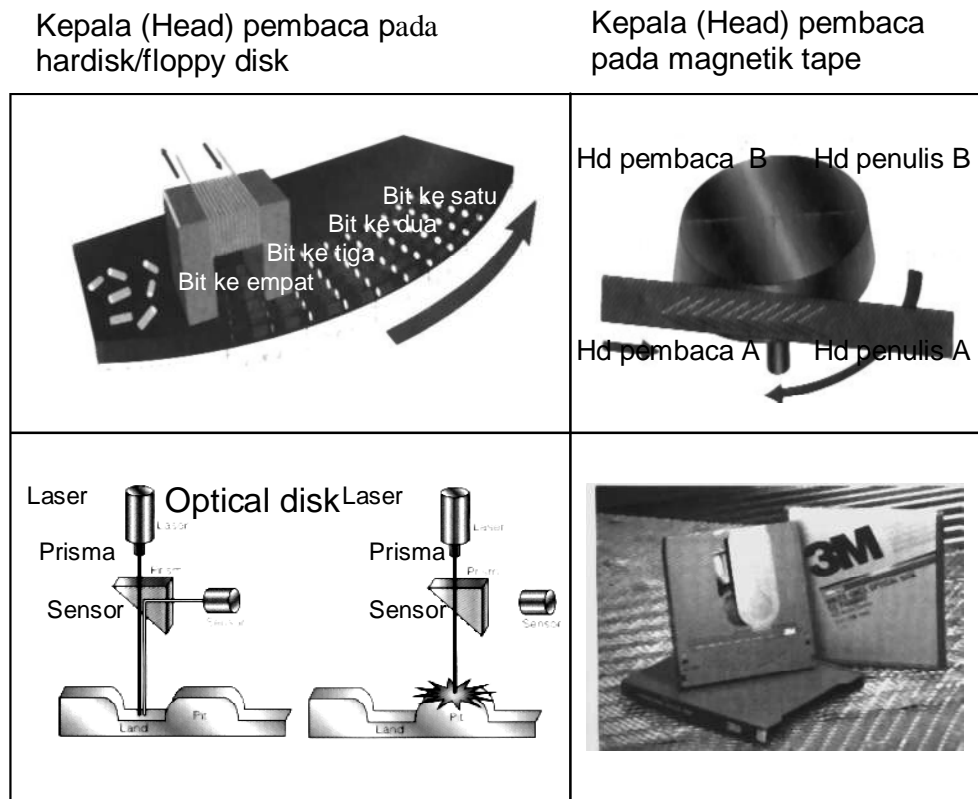
- Terkontribusi** - maksudnya adalah bahwa setiap bagian dari *data* dalam *database* dapat memberikan kontribusi (*shared*) kepada pemakai yang berbeda pada waktu yang sama sesuai dengan hak akses yang diberikan

### 12.4.2. Komponen Hardware

Bagian hardware dari sistem *database* meliputi :

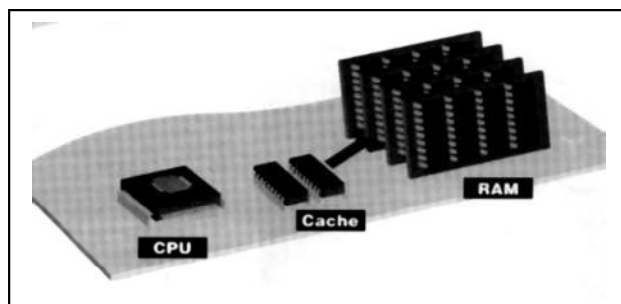
- Kepala (*head*) pembaca** - yang digunakan untuk mengambil dan membaca *data* bersama-sama dengan bagian *I/O (input Output), controler, Kabel I/O, disk* dan sebagainya.

Gambar 12.3 Beberapa macam head dan disk



- ❑ **Prosesor (Processor) dan Memori** - yang digunakan untuk mendukung jalannya *software* sistem *database*.

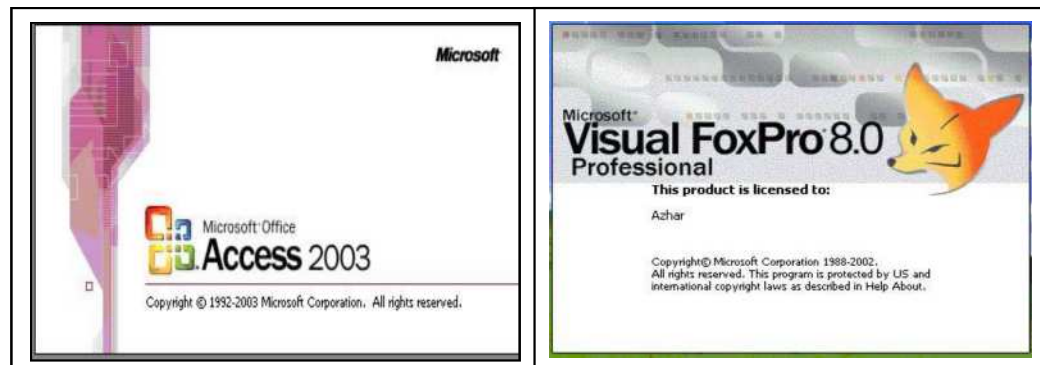
Gambar 12.4 Memori dan prosesor



### 12.4.3. Komponen Software

Antara *database* secara fisik (*Data* yang tersimpan dalam *disk* secara fisik) dan pemakai sistem informasi manajemen dibatasi oleh suatu *software* (*Database manager*) atau lebih umum dikenal sebagai Sistem Manajemen *Database* (*DBMS*). Semua kebutuhan pemakai untuk mengakses *database* ditangani oleh *DBMS*. *DBMS* juga dapat digunakan untuk menyusun suatu aplikasi sistem informasi manajemen perusahaan yang siap pakai sehingga *user* tidak perlu lagi menghapalkan perintah-perintah seperti halnya dalam *DBMS*.

Gambar 12.5 Beberapa software DBMS generasi baru

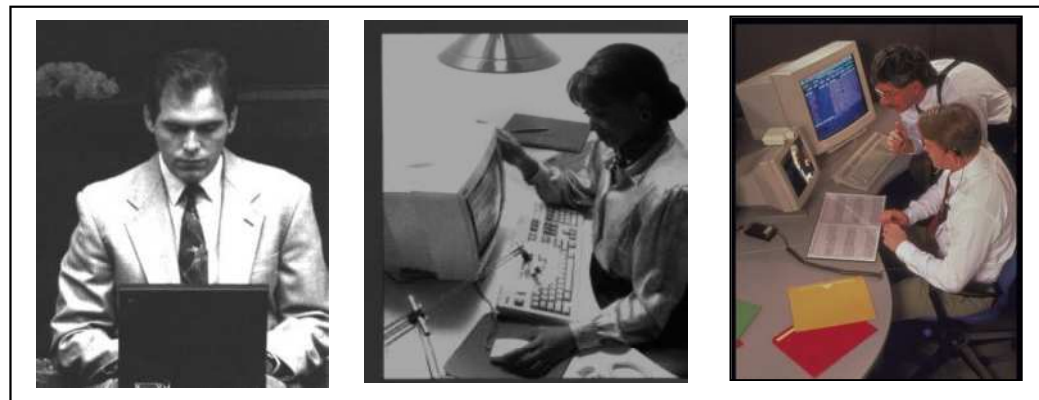


#### 12.4.4. Komponen Pemakai

Dalam sistem *data base* ada tiga kelas pemakai, yaitu :

1. Kelompok pertama adalah *Programer* aplikasi yang bertanggung jawab dalam membuat program aplikasi yang menggunakan *database*
2. Kelompok kedua adalah pemakai (*end user*) yang berinteraksi dengan *system* melalui terminal komputer yang *online*
3. Kelompok ketiga adalah administrator database (DBA)

Gambar 12.6 Programmer , pemakai akhir (operator) dan database administrator



### 12.5 Abstraksi Data

Abstraksi *data* digunakan dalam struktur *database* modern, didalamnya terdapat tiga level abstraksi yang dikenal sebagai: internal, konseptual dan eksternal Seperti terlihat pada gambar 12.5 di bawah ini

- ❑ **Tingkat internal** - tingkat ini merupakan tingkat yang paling dekat dengan media penyimpanan secara fisik. Tingkat ini lebih menjelaskan secara detail bagaimana seharusnya *data* secara fisik disimpan dalam media penyimpanan.

- ❑ **Tingkat eksternal** - tingkat ini merupakan tingkat yang paling dekat kepada pemakai (*User*).Tingkat ini berhubungan dengan bagaimana *data* dipandang oleh pemakai. Dalam kenyatannya *data* yang ada pada tingkat ini merupakan *data-data* yang dapat kita lihat pada saat kita memasukan *data* ke terminal komputer. Dan perlu diketahui disini bahwa tingkat ini hanya menggambarkan sebagian saja dari *data base*.
- ❑ **Tingkat konseptual** - tingkat ini menggambarkan *data* apa yang sebenarnya disimpan dalam database dan hubungan yang terjadi diantara *data-data*. Disini Keseluruhan database digambarkan dalam bentuk struktur yang relatif sederhana. Meskipun menggunakan struktur yang sederhana pada tingkat konseptual dilakukan akan tetapi tetap saja struktur tersebut cukup kompleks bila menggunakan database yang besar

**Tiga tingkatan abstraksi data:**

- Internal
- Konseptual
- Eksternal

Gambar 12.7 Tiga tingkat abstraksi data

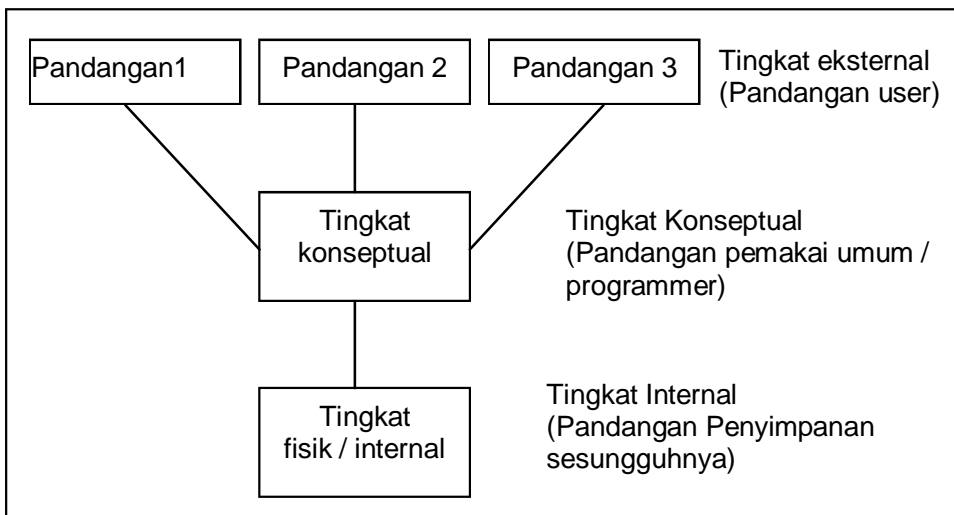


Table 12.12 Contoh Tiga Tingkat Abstraksi Data

Eksternal	Internal	Konseptual
<p><b>Foxpro :</b>                      @ 8,8 Say EMP_NUM                      @11,8 Say MP_NAME</p>	<p>Stored_EMP Length = 18                      Prefix Type=Byte(6),                      Offset = 0                      EMP# Type=Byte(6),                      Offset = 6,                      Index= EMPX                      EMP Type=Byte(20),                      Offset = 12</p>	<p>Create Table Employe                      EMP_NUM Character (6)                      EMP_NameCharacter (20)</p>

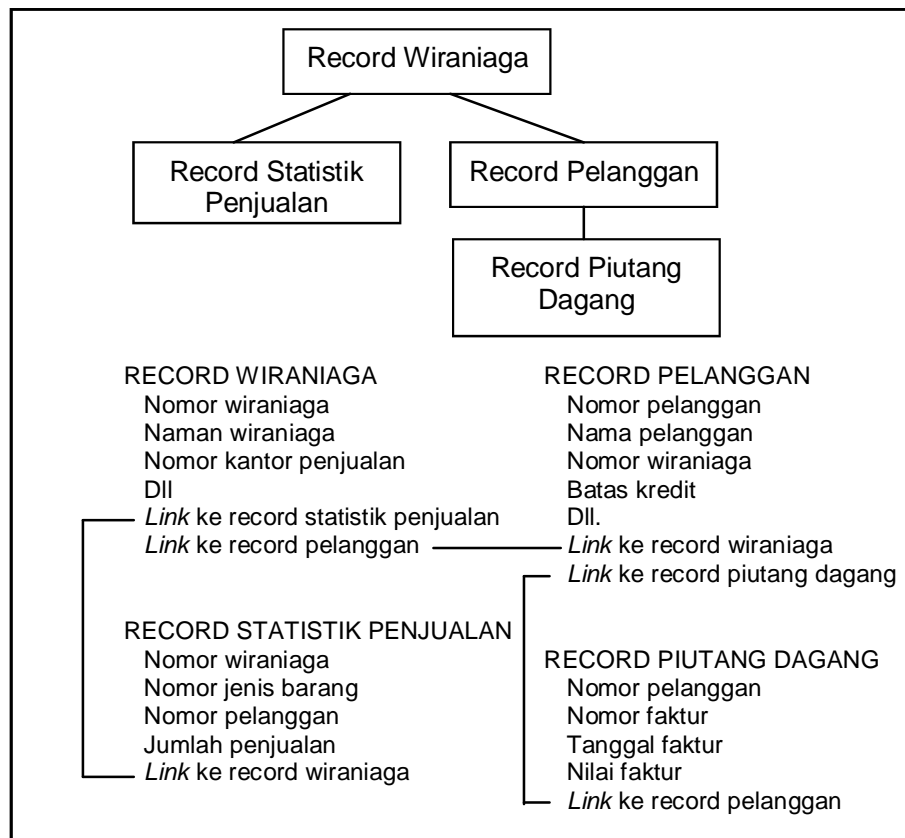
## 12.6 Model-Model Organisasi Data

Dasar penyusunan struktur sebuah database adalah berdasarkan kepada model *data* yang digunakan, model *data* ini merupakan kumpulan dari alat-alat atau lambang-lambang yang digunakan untuk menggambarkan *data* secara konseptual. Saat ini ada beberapa model *data* yang sering digunakan seperti diantara para pengembang sistem informasi manajemen misalnya:

- ❑ **Model Hirarki (*Hierarchical data model*)** - Model *data* yang menggambarkan hubungan antara *data* berdasarkan kepada tingkatannya.
- ❑ **Model Network (*Network data model*)** - Model *data* yang menggambarkan hubungan antar *data* berdasarkan kepentingannya.
- ❑ **Model Relasi (*Relational data model*)** - Model *data* yang disusun berdasarkan kepada hubungan antar dua entitas (*entity*).

### 12.6.1. Model Hirarki

Gambar 12.8 Model *data* secara hirarki



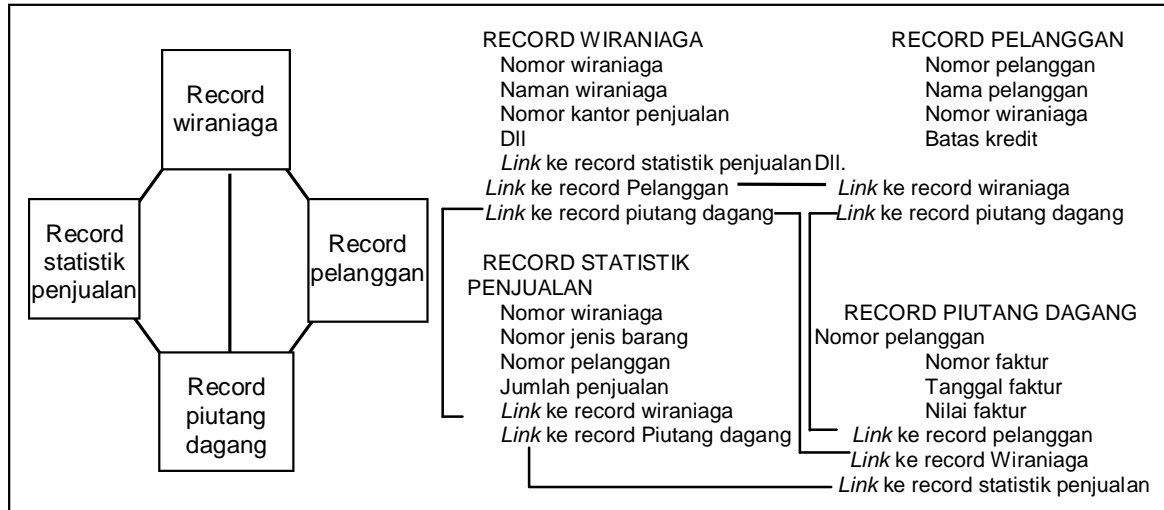
Model *data* secara hirarki seperti gambar 12.8 diatas juga dikenal sebagai model pohon, Pada model *data* ini *data* induk memiliki beberapa percabangan sesuai dengan kebutuhan dan



masing-masing cabang juga memiliki cabang lain sehingga kalau digambarkan akan berbentuk seperti struktur organisasi. Model *data* ini merupakan salah satu model lama yang masih banyak digunakan.

### 12.6.2. Model Network

Gambar 12.9. Model Network



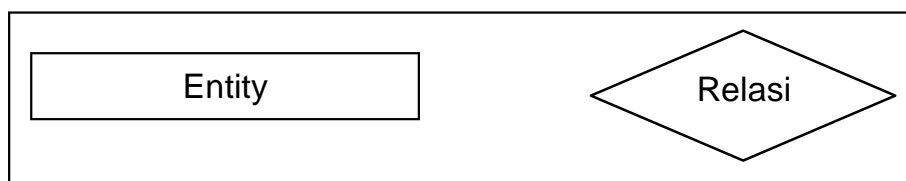
Model *Network* diatas merupakan perkembangan lebih lanjut dari model *data* secara hirarki. Pada model ini setiap *file data* dapat berhubungan dengan *file-file data* lainnya sesuai dengan kebutuhan manajemen suatu organisasi perusahaan. Dengan adanya hubungan langsung dari satu *file data* ke *file data* yang lain secara langsung tanpa harus melewati dahulu satu atau beberapa *file data* maka akses *data* akan lebih cepat.

### 12.6.3. Model Relasi

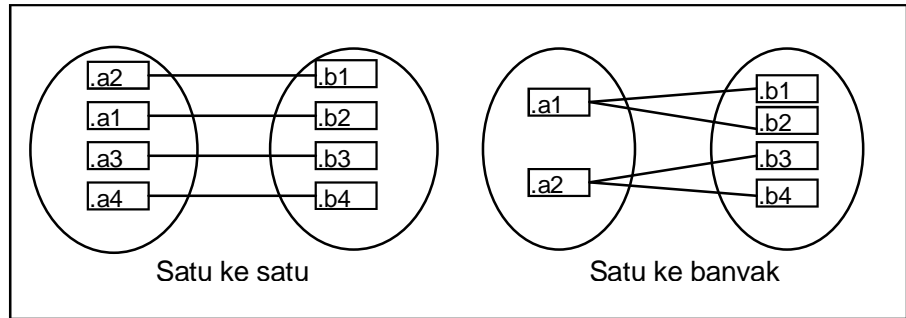
Model relasi didasarkan kepada persepsi tentang dunia nyata yang berisi sekumpulan objek-objek dasar yang disebut sebagai *entity* dan hubungan antara *entity-entity* tersebut. Beberapa cara penggambaran salah satu model relasi yaitu model relasi Peter Chen serta variasi perkembangannya.

Gambar 12.10 berikut ini menunjukkan lambang-lambang yang digunakan dalam model relasi Peter Chen.

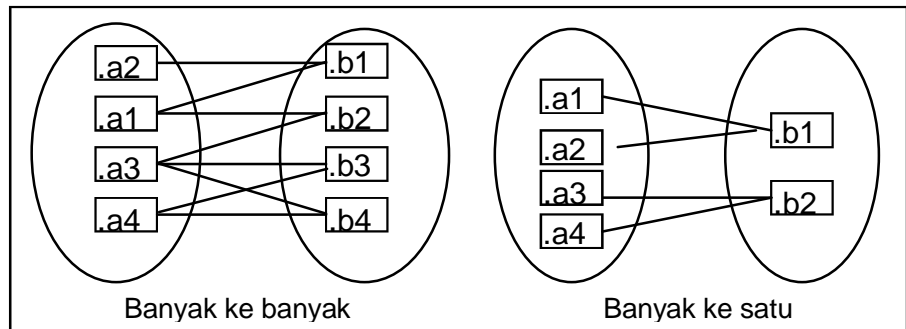
Gambar 12.10 Lambang dalam model relasi Peter chen



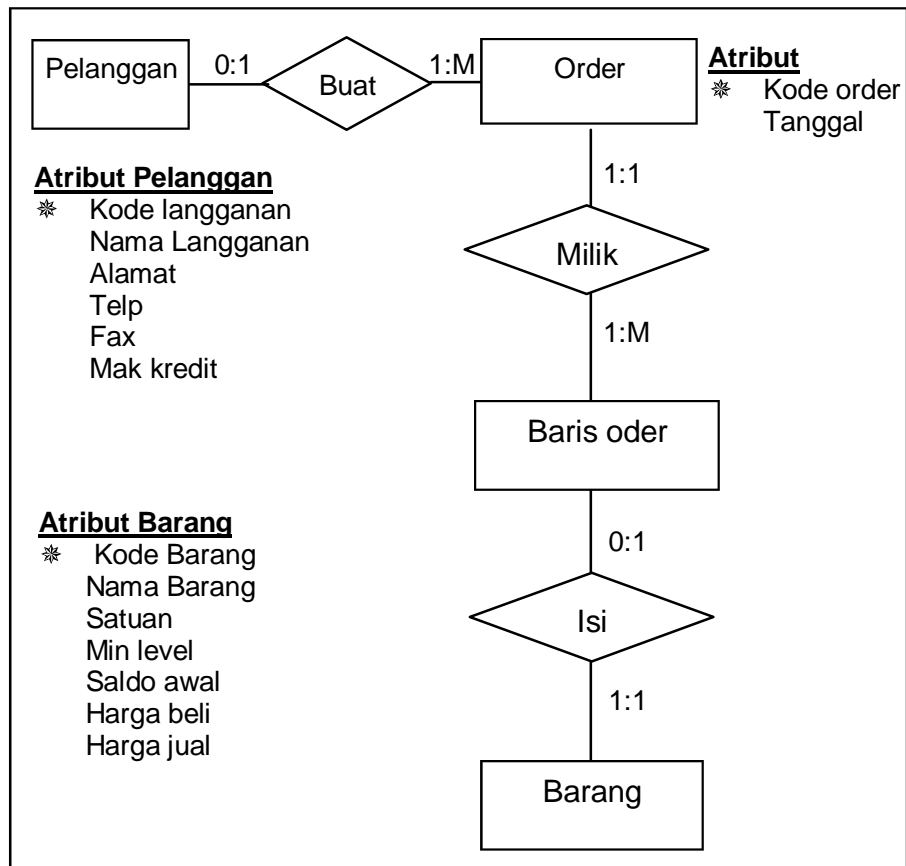
Gambar 12.11 Tingkat hubungan satu ke satu dan satu ke banyak



Gambar 12.12 Tingkat hubungan banyak ke banyak dan banyak ke satu

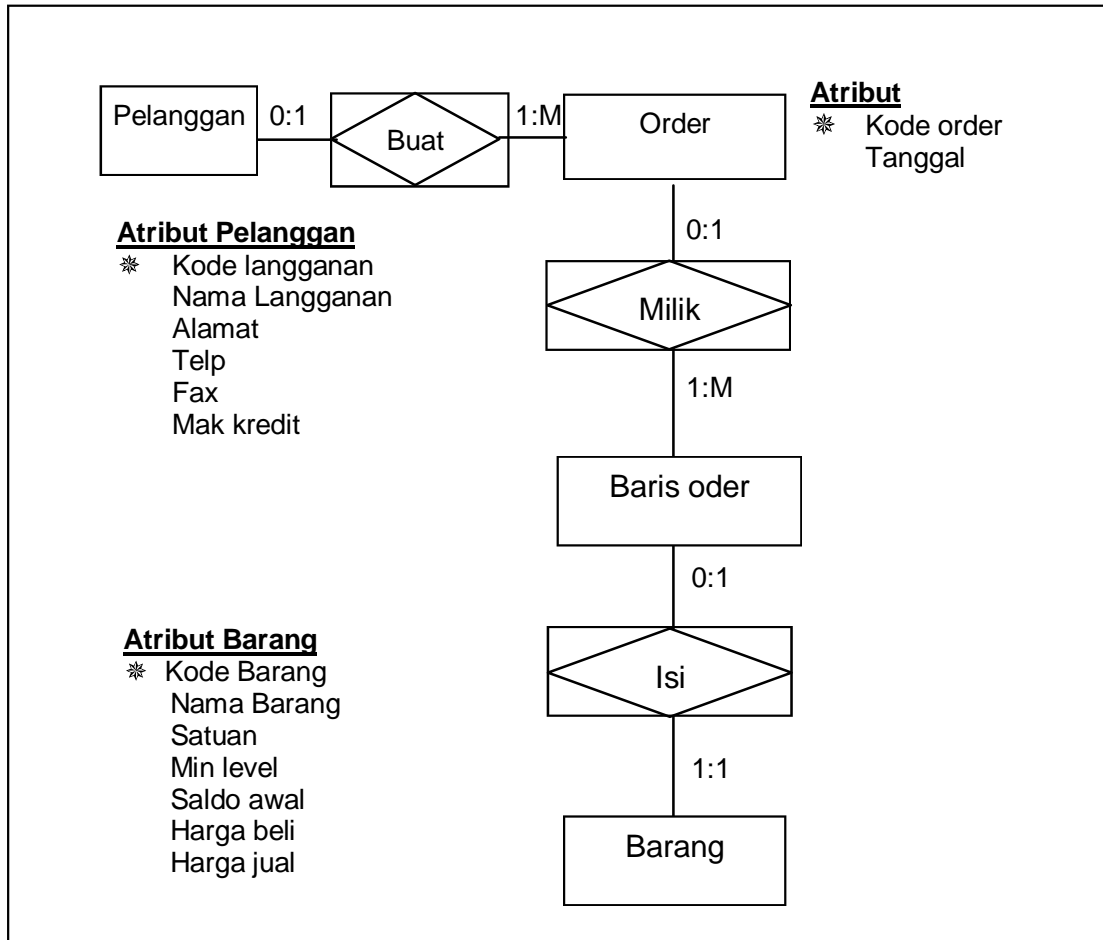


Gambar 12.13 Model ERD Peter Chen



Perkembangan lainnya didalam pembuatan diagram ERD Peter Chen disarankan agar suatu relasi juga berfungsi sebagai suatu entity (model ini sangat banyak disukai), sehingga modelnya dapat digambarkan seperti pada halaman berikut:

Gambar 12.14. Pengembangan ERD model Peter Chen dimana relasi sebagai entitas (entity)



### 12.6.4. Perangkat Lunak Database

Banyak perangkat lunak yang digunakan untuk mengelola dalam *data database*. Beberapa perangkat lunak yang terkenal dan beredar di Indonesia saat ini seperti:

- Foxpro, Visual Foxpro
- Clipper, Dbase
- Oracle, SQL for windows, DB2, Ingres

Perangkat-perangkat lunak tersebut memiliki cara yang tidak persis sama dalam menggambarkan *entity* dan relasinya dalam *database environment* halaman berikut ini contoh dari *entity* yang ditunjukkan dalam Visual FoxPro serta bagaimana mereka dihubungkan atau direlasikan.

## Rangkuman

---

*Data* yang masuk dalam sistem informasi tidak semuanya langsung diolah (*on-line*) tapi ada juga yang disimpan dulu dan diolah nanti saat diperlukan (*Batch*), dalam menyimpan *data*, *data* disusun berdasarkan susunan tertentu secara tradisional penyusunan *data* lebih menekankan kepada bagaimana *data* tersebut disimpan didalam media penyimpanan sedangkan secara modern, penyusunan *data* didalam media penyimpanan *data* merupakan bagian dari konsep manajemen *data*. Dalam konsep manajemen *data*, *data* disusun secara hirarki, *network* atau berdasarkan relasi.

### Soal

1. Apa yang dimaksud dengan *database* dan jelaskan pengertian *database* secara luas dan sempit ?
2. Coba jelaskan dan beri contoh tentang hirarki *data* ?
3. Sebutkan berbagai cara menentukan alamat *record* untuk menyimpan *data* ?
4. Jelaskan masalah dalam organisasi/susunan *data* tradisional
5. Sebutkan dan jelaskan kegiatan manajemen *data* dan organisasi *datanya* ?

### Tugas

1. Coba gambarkan dengan model hirarki *data-data* yang berhubungan dengan penjualan pada perusahaan dagang kebutuhan pokok (grosir).
2. Coba gambarkan dengan model *network data-data* yang berhubungan dengan perusahaan dagang obat-obatan seperti apotik/grosir obat.
3. Coba gambarkan dengan model relasi *data-data* penjualan pada perusahaan dagang onerdil mobil.
4. Mengapa dalam organisasi *data* tradisional *data* yang disimpan susah dibaca dan kurang aman ?
5. Mengapa sistem organisasi *data* modern dapat mengatasi sistem organisasi *data* tradisional ?