

Pengantar Database

Faried Irmansyah

fariedir@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Data

Nilai/*value* yang turut merepresentasikan deskripsi dari suatu objek atau kejadian (event)

Informasi

Merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.

Sistem Informasi

Suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan.

Sistem informasi = Sistem terotomasi, terdiri dari beberapa komponen, antara lain :

- *Hardware* : CPU, Disk, terminal, printer.
- *Software* : Sistem operasi, sistem database, prog. pengontrol komunikasi, prog. aplikasi
- *Personil* : yg mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem.
- *Data* : data yang tersimpan dalam jangka waktu tertentu
- *Prosedur* : instruksi dan kebijakan untuk mengoperasikan sistem

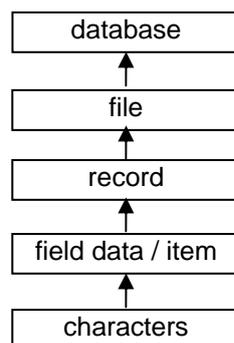
Database

Kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu.

Mengapa Diperlukan Database

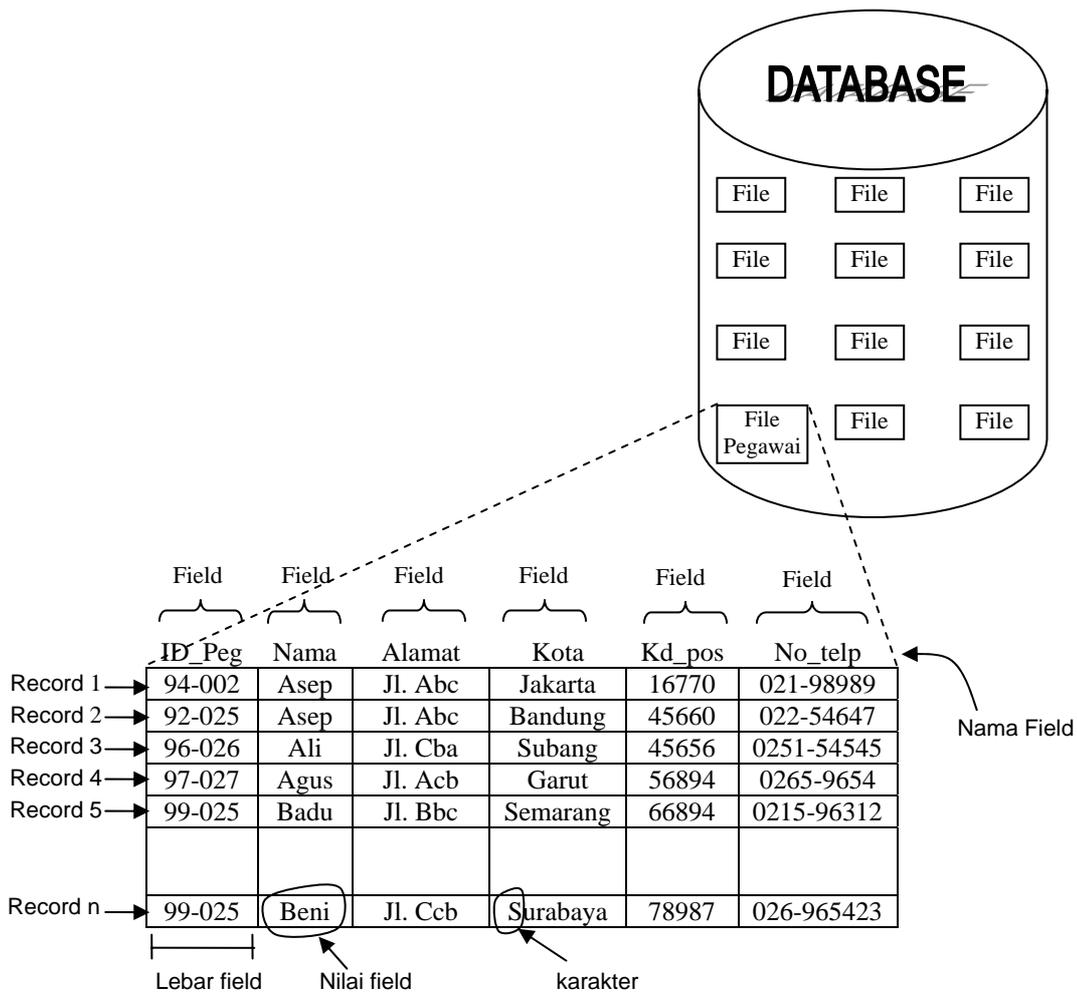
- Salah satu komponen penting dalam *sistem informasi*, karena merupakan dasar dalam menyediakan informasi
- Menentukan kualitas informasi : akurat, tepat pada waktunya dan relevan. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- Mengurangi duplikasi data (data redundancy)
- Hubungan data dapat ditingkatkan (data relatability)
- Mengurangi pemborosan tempat simpanan luar

Jenjang Data



- ▶ **Characters** : merupakan bagian data yang terkecil, dapat berupa karakter numerik, huruf ataupun karakter-karakter khusus (special characters) yang membentuk suatu item data / field.
- ▶ **Field** : merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari field membentuk suatu record.
 - field name: harus diberi nama untuk membedakan field yang satu dengan lainnya
 - field representation: tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan karakter-karakter data).
 - field value: isi dari field untuk masing-masing record.
- ▶ **Record** : Kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap-tiap record dapat mewakili data tiap-tiap karyawan.

- ▶ **File:** File terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis. Misalnya file mata pelajaran berisi data tentang semua mata pelajaran yang ada.
- ▶ **Database :** Kumpulan dari file / tabel membentuk suatu database



Tipe File

1. File Induk (*master File*)
 - a. file induk acuan (*reference master file*) : file induk yang recordnya relatif statis, jarang berubah nilainya. Misalnya file daftar gaji, file mata pelajaran.
 - b. file induk dinamik (*dynamic master file*): file induk yang nilai dari record-recordnya sering berubah atau sering dimutakhirkan (*update*) sebagai hasil dari suatu transaksi. Misalnya file induk data barang, yang setiap saat harus di *up-date* bila terjadi transaksi.

2. File Transaksi (*transaction file*)
File ini bisa disebut *file input*; digunakan untuk merekam data hasil dari transaksi yang terjadi. Misalnya file penjualan yang berisi data hasil transaksi penjualan.
3. File Laporan (*Report file*)
File ini bisa disebut *output file*, yaitu file yang berisi informasi yang akan ditampilkan.
4. File Sejarah (*history file*)
File ini bisa disebut file arsip (*archival file*), merupakan file yang berisi data masa lalu yang sudah tidak aktif lagi, tetapi masih disimpan sebagai arsip.
5. File Pelindung (*backup file*)
File ini merupakan salinan dari file-file yang masih aktif di dalam database pada suatu saat tertentu. File ini digunakan sebagai pelindung atau cadangan bila file database yang aktif mengalami kerusakan atau hilang.

Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).

Normal Pertama (1st Normal Form)

Aturan :

- Mendefinisikan atribut kunci
- Tidak adanya group berulang
- Semua atribut bukan kunci tergantung pada atribut kunci

Normalisasi Kedua (2nd Normal Form)

Aturan :

- Sudah memenuhi dalam bentuk normal kesatu
- Sudah tidak ada ketergantungan parsial, dimana seluruh field hanya tergantung pada sebagian field kunci.

Normalisasi Ketiga (3rd Normal Form)

Aturan :

- Sudah berada dalam bentuk normal kedua
- Tidak ada ketergantungan transitif (dimana field bukan kunci tergantung pada field bukan kunci lainnya).

Catatan:

Normal seharusnya berada dalam bentuk normal tertinggi dan bergerak dari bentuk normal satu dan seterusnya untuk setiap kali membatasi hanya satu jenis redundansi. Keseluruhannya cuma ada lima bentuk normal. Tiga bentuk normal pertama menekankan redundansi yang muncul dari *Function Dependencies* sedangkan bentuk

keempat dan kelima menekankan redundansi yang muncul dari kasus *Multi Valued Dependencies*.

Contoh 1

relasi tidak normal (Tabel Order)

No Order	Tanggal Order	Isi Order	Jumlah Item
A20	5 Juli 1987	No Item	
		Pc6	24
		Bw3	83
D33	20 Mei 1997	Ty6	37
		No Item	
		Pc5	89
		Bw3	15
D33	20 Mei 1997	Ty6	33
		Hj7	45

Relasi normal (Tabel Personil)

No Order	Tanggal Order	Isi Order	Jumlah Item
A20	5 Juli 1987	Pc6	24
A20	5 Juli 1987	Bw3	83
A20	5 Juli 1987	Ty6	37
D33	20 Mei 1997	Pc5	89
D33	20 Mei 1997	Bw3	15
D33	20 Mei 1997	Ty6	33
D33	20 Mei 1997	Hj7	45

Contoh 2

relasi tidak normal (Tabel Personil)

ID_Personil	Tanggal Lahir	Karakteristik
I102	17 Januari 1970	Tinggi 162
		Berat 50
		Rambut hitam
A212	12 Desember 1966	Tinggi 170
		Berat 64
		Rambut hitam

Relasi Normal (Tabel Personil)

ID Personil	Tanggal lahir	Tinggi	Berat	Warna Rambut
I102	17 Januari 1970	162	50	Hitam
A212	12 Desember 1966	170	64	Hitam

Contoh 3 : Langkah-langkah normalisasi

Tabel data yang mendeskripsikan karyawan yang bekerja pada proyeknya.

Project number	Project name	Employee number	Employee name	Rate category	Hourly rate
1023	Madagascar travel site	11	Vincent Radebe	A	\$60
		12	Pauline James	B	\$50
		16	Charles Ramoraz	C	\$40
1056	Online estate agency	11	Vincent Radebe	A	\$60
		17	Monique Williams	B	\$50

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama.

Field-field tabel di atas yang merupakan group berulang : Employee number, Employee name, Rate category, Hourly rate.

Normalisasi Pertama :

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial; menjadikan field-field menjadi tergantung pada satu atau beberapa field, bukan seluruhnya.

Karena yang dapat dijadikan kunci adalah *Project Number* dan *Employee Number*, maka langkah kemudian dicari field-field mana yang tergantung pada *Project Number* dan mana yang tergantung pada *Employee Number*.

employee_project table

<i>Project number</i>	Project name	<i>Employee number</i>	Employee name	Rate category	Hourly rate
1023	Madagascar travel site	11	Vincent Radebe	A	\$60
1023	Madagascar travel site	12	Pauline James	B	\$50
1023	Madagascat travel site	16	Charles Ramoraz	C	\$40
1056	Online estate agency	11	Vincent Radebe	A	\$60
1056	Online estate agency	17	Monique Williams	B	\$50

Ket : nama field yang dicetak miring dan tebal menunjukkan primary key.

Ada beberapa keanehan dalam data pada tabel di atas. Adanya kesalahan penulisan data pada kolom project name record ketiga. Kemungkinan kesalahan penulisan data semakin bertambah seiring bertambahnya record.

Normalisasi Kedua :

Field-field yang tergantung pada satu field haruslah dipisah dengan tepat, misalnya *Project Number* menjelaskan *Project Name* dan *Employee Number* menjelaskan *Employee Name*, *Rate Category* dan *Hourly Rate*.

Untuk membuat hubungan antara dua tabel, dibuat suatu tabel yang berisi key-key dari tabel yang lain.

employee_project table

<i>Project number</i>	<i>Employee number</i>
1023	11
1023	12
1023	16
1056	11
1056	17

Employee table

<i>Employee number</i>	Employee name	Rate category	Hourly rate
11	Vincent Radebe	A	\$60
12	Pauline James	B	\$50
16	Charles Ramoraz	C	\$40
17	Monique Williams	B	\$40

Project table

<i>Project number</i>	Project name
1023	Madagascar travel site
1056	Online estate agency

Pada tabel Project dapat dilihat bahwa Project Name hanya disimpan sekali. Secara jelas bahwa pada tabel employee project tabel mengisi data yang berulang-ulang. Tapi paling tidak lebih efisien dan mengurangi resiko kesalahan dalam penulisan data yang panjang. Database di atas masih belum sempurna. Masih ada anomali pada data.

Normalisasi Ketiga :

Pada tabel diatas masih terdapat masalah, bahwa *Employee Number* 17 mendapatkan hourly rate yang tidak sesuai dengan *rate category* nya.

Solusinya adalah kita harus mencari hubungan transitif (*transitive relation*) dimana field non-key tergantung pada field non-key lainnya.

Artinya kita harus memisahkan field non-kunci *Rate Category* yang tadinya tergantung secara parsial kepada field kunci *Employee Number*, untuk menghilangkan anomali penulisan data pada field hourly rate.

Employee_project table

<i>Project number</i>	<i>Employee number</i>
1023	11
1023	12
1023	16
1056	11
1056	17

Employee table

<i>Employee number</i>	Employee name	Rate category
11	Vincent Radebe	A
12	Pauline James	B
16	Charles Ramoraz	C
17	Monique Williams	B

Rate table

<i>Rate category</i>	Hourly rate
A	\$60
B	\$50
C	\$40

Project table

<i>Project number</i>	Project name
1023	Madagascar travel site
1056	Online estate agency

Tabel-tabel yang memenuhi kriteria normalisasi ketiga, sudah siap diimplementasikan. Sebenarnya masih ada lagi bentuk normalisasi yang lain; Normalisasi Boyce-Codd dan normalisasi keempat, hanya saja sangat jarang dipakai. Pada kebanyakan kasus, normalisasi hanya sampai ketiga.

Entity Relationship Diagram (ERD)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. Dan dengan ERD kita mencoba menjawab pertanyaan seperti; data apa yang kita perlukan? bagaimana data yang satu berhubungan dengan yang lain?

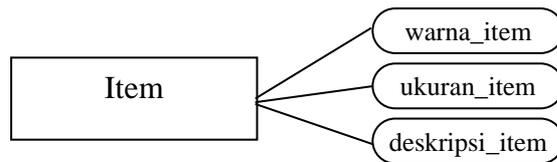
ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan yaitu :

1. **Entiti** : adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja dan lain-lain. Seandainya A adalah seorang pekerja maka A adalah isi dari pekerja, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan. Karena itu harus dibedakan antara

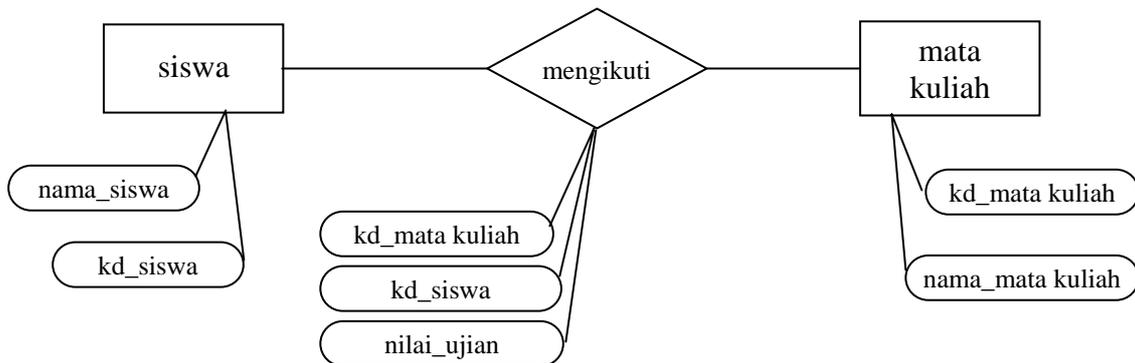
entiti sebagai bentuk umum dari deskripsi tertentu dan isi entiti seperti A dan B dalam contoh di atas. Entiti digambarkan dalam bentuk persegi empat.



2. **Atribut** : Entiti mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja. Setiap ERD bisa terdapat lebih dari satu atribut. Entiti digambarkan dalam bentuk ellips.



3. **Hubungan** : Relationship; sebagaimana halnya entiti maka dalam hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri. Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai_ujian. Relationship digambarkan dalam bentuk intan / diamonds.

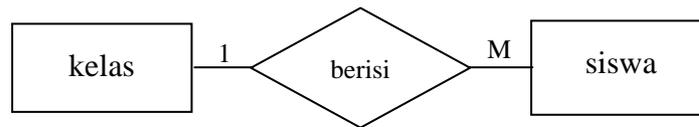


Jenis-jenis hubungan :

- **satu ke satu**, misalnya dalam suatu perusahaan mempunyai aturan satu supir hanya boleh menangani satu kendaraan karena alasan tertentu.



- **satu ke banyak** / banyak ke satu, misalnya suatu sekolah selalu mempunyai asumsi bahwa satu kelas terdiri dari banyak siswa tetapi tidak sebaliknya, yaitu satu siswa tidak dapat belajar pada kelas yang berbeda.



ERD dapat digambarkan lengkap dengan atribut-atributnya, bisa juga digambarkan tanpa atributnya.

Relasi (Relational Database Management System / RDBMS)

Merupakan sekumpulan data yang disimpan sedemikian rupa sehingga mudah diambil informasinya bagi pengguna, dan data tersebut saling berhubungan.

RDBMS merupakan suatu paket perangkat lunak yang kompleks digunakan untuk memanipulasi database.

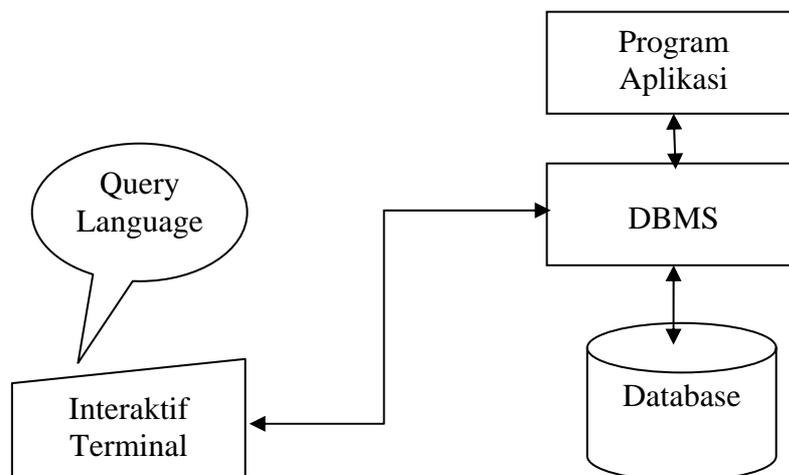
Ada tiga prinsip dalam RDBMS :

- Data definition
Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara relasi data, validasi data dan lainnya.
- Data Manipulation
Data yang telah dibuat dan didefinisikan tersebut akan dilakukan beberapa pengerjaan, seperti menyaring data, melakukan proses *query*, dsb
- Data Control
Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak user, dsb

Semua operasi input dan output yang berhubungan dengan database harus menggunakan DBMS. Bila pemakai akan mengakses database, DBMS menyediakan penghubung (interface) antara pemakai dengan database.

Hubungan pemakai dengan database dapat dilakukan dengan dua cara :

1. Secara interaktif menggunakan bahasa pertanyaan (query language).
2. Dengan menggunakan program aplikasi.



Beberapa Software Database :

Dbase, Foxbase, Foxpro, Microsoft Access, File Maker Pro, DB2, Postgres/Ingres, Microsoft SQL Server, Oracle, Power Builder, Sybase, MySQL, dsb

SQL

SQL (dibaca "ess-que-el") singkatan dari Structured Query Language. SQL adalah bahasa yang digunakan untuk berkomunikasi dengan database. Menurut ANSI (American National Standards Institute), bahasa ini merupakan standard untuk relational database management systems (RDBMS).

Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti : update data pada database, atau menampilkan data dari database. Beberapa software RDBMS dan dapat menggunakan SQL, seperti : **Oracle, Sybase, Microsoft SQL Server, Microsoft Access, Ingres**, dsb. Setiap software database mempunyai bahasa perintah / sintaks yang berbeda, namun pada prinsipnya mempunyai arti dan fungsi yang sama.

Perintah-perintah tsb antara lain : "Select", "Insert", "Update", "Delete", "Create", dan "Drop", yang dapat digunakan untuk mengerjakan hampir semua kebutuhan untuk memanipulasi sebuah database.

Di bawah ini adalah beberapa contoh perintah-perintah Query dalam Microsoft SQL Server :

Create : digunakan untuk membuat tabel baru

```
create table "tablename" ("column1" "data type" [constraint], "column2" "data type" [constraint], "column3" "data type" [constraint]);  
[ ] = optional
```

contoh :

```
create table empinfo (Id varchar (5), first_name varchar(15), last_name varchar(20), address varchar(30), city varchar(20), state varchar(20));
```

hasil dari perintah di atas :

Id	first_name	last_name	address	city	state
----	------------	-----------	---------	------	-------

Select : digunakan untuk menampilkan data sesuai kriteria yang kita tentukan

```
select "column1" [,"column2",etc] from "tablename" [where "condition"];  
[ ] = optional
```

Misalnya sebuah tabel yang sudah terisi beberapa record seperti di bawah ini :

Table: empinfo

Id	first_name	last_name	address	city	state
----	------------	-----------	---------	------	-------

99980	John	Jones	13 th south street	Payson	Arizona
99982	Mary	Jones	Malibu Street	Payson	Arizona
88232	Eric	Edwards	Jampang 212	San Diego	California
88233	Mary Ann	Edwards	Jl. Candraloka 14	Phoenix	Arizona
98002	Ginger	Howell	Kemang 67	Cottonwood	Arizona
92001	Sebastian	Smith	Parung 111	Gila Bend	Arizona
22322	Gus	Gray	Beverly 21	Bagdad	Arizona
32326	Mary Ann	May	Jl. Sudirman 10	Tucson	Arizona
32327	Erica	Williams	Jl. Merdeka 45	Show Low	Arizona
32380	Leroy	Brown	Basrah street 03	Pinetop	Arizona
32382	Elroy	Cleaver	Jl. Kahuripan 34	Globe	Arizona

Tampilkan first name dan address untuk tabel di atas :

select first_name, age **from** empinfo;

Tampilkan first name, last name, and city yang bukan Payson.

select first_name, last_name, city **from** empinfo **where** city <> 'Payson';

Tampilkan semua kolom dimana first name sama dengan "Mary".

select * from empinfo **where** first_name = 'Mary';

Tampilkan semua kolom dimana first name mengandung "Mary".

select * from empinfo **where** first_name **LIKE** '%Mary%';

Insert : digunakan untuk menyisipkan atau menambah baris pada tabel.

insert into "tablename" (first_column,...last_column)

values (first_value,...last_value);

contoh :

Sisipkan ke dalam table empinfo dengan pada field **first name** : Luke, **last name** : Duke, **address** :2130 Boars Nest, **city** : Peachtree, **State** : Georgia

Perintahnya adalah sbb :

insert into empinfo (first_name, last_name, address, city, state)

values ('Luke', 'Duke', '2130 Boars Nest', 'Peachtree', 'Georgia');

Update : digunakan untuk mengupdate atau merubah isi data dalam tabel

update "tablename" set "columnname" = "newvalue" [,"nextcolumn" =

"newvalue2"...] where "columnname" OPERATOR "value"

[and/or "column" OPERATOR "value"];

[] = optional

contoh :

Perhatikan tabel di atas; Pegawai yang bernama Elroy Cleaver pindah rumah dengan alamat yang baru yaitu; address : Jl.Bojong 12, city : Depok, State : West Java.

Perintahnya adalah sbb :

```
update empinfo set address = 'Jl.Bojong 12', city = 'Depok', State = 'West Java'  
where id=32382
```

Delete : digunakan untuk menghapus baris/record data dalam tabel
delete from "tablename" where "columnname" OPERATOR "value"
[and/or "column" OPERATOR "value"];

[] = optional

contoh :

```
delete from empinfo where lastname = 'May';
```

Drop : digunakan untuk menghapus tabel
drop table "tablename"

contoh :

```
drop table empinfo
```

Daftar Pustaka

Pengantar Perancangan Sistem, Erlangga
Jogiyanto, Pengenalan Komputer, 1995
<http://www.sqlcourse.com>