BAB 3
BAHASA BASIS DATA
(DATABASE LANGUAGE)

DBMS merupakan perantara bagi pemakai dengan basis data dalam Disk. Cara berkomunkasi / berinteraksi antara pemakai dengan basis data diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa ini disebut sebagai bahasa basis data yang meliputi sejumlah perintah (statement) yang diformulasikan dan dapat diberikan user dan dikenali / diproses DBMS untuk melakukan suatu aksi / pekerjaan tertentu.

3.1. Deskripsi Bahasa Basisdata

Bahasa basis data umumnya dapat ditempelkan (embbeded) ke bahasa pemrogarman lain, misalkan ditempelkan kedalam bahasa Java,C/C++, Pascal, Basic, Fortran, Ada dan lainnya. Bahasa tempat ditempelkannya instruksi bahasa basisdata disebut sebagai inang (*host language*). Pada program bahasa inag yang ditempeli kode-kode bahasa basisdata mka saat *source program* dikompilasi maka *source program* terlebih dahulu dilewatkan ke pre-kompilator (*pre-compiler*) yang menterjemahkan instruksi bahasa basisdata menjadi instruksi asli bahasa pemrograman inang.

Setelah itu, kode-kode yang telah dalam bahasa inang dikompilaasi menggunakan kompilator bahasa inang sehingga menghasilkan kode biner yang mengaitkan (*link*) instruksi – instruksi bahasa basisdata ke pustaka *(library*) yang berfungsi menghubung DBMS. Hasil – hasil instruksi bahasa basisdata segera ditangkap program bahasa inang dan diolah sesuai keperluan aplikasi.

Selain itu vendor DBMS juga selalu menyertakan fasilitas inteaktif memberi perintah ke DBMS secara langsung. Cara ini umumnya dipergunakan administrator untuk menjalankan tugasnya mengolah seluruh basisdata organisasi.

3.2. Komponen Bahasa Basisdata

Berdasarkan fungsinya, bahasa basis data dapat dipilah ke dalam 3 (tiga) bentuk yaitu :

1. *Data Definition Language* (DDL)
2. *Data Manipulation Language* (DML)
3. *Data Control Language* (DCL)

DDL berfungsi menspesifikasikan skema atau struktur basisdata, hasil pernyataan DDL adalah himpunan definisi data yang disimpan secara khusus pada *data dictionary* (*data directory*).

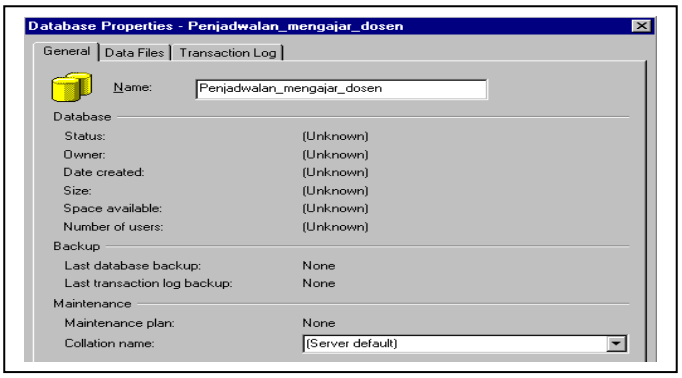
DML berisi sekumpulan operasi manipulasi data pada basisdata, DML biasa disebut bahasa *query* yaitu bahasa untuk meminta informasi dari basisdata karena komponen paling kompleks di DML adalah operaasi *query*. Sebenarnya DML tidak hanya berisi operasi utnuk *query*, namun juga meliputi operasi penghapusan, pembaruan dan penyisipan.

DCL merupakan sub bahasa untuk mengendalikan struktur internal basisdata, DCL untuk menyesuaikan sistem agar supaya lebih efisian dan DCL sangat bergantung pada vendor.

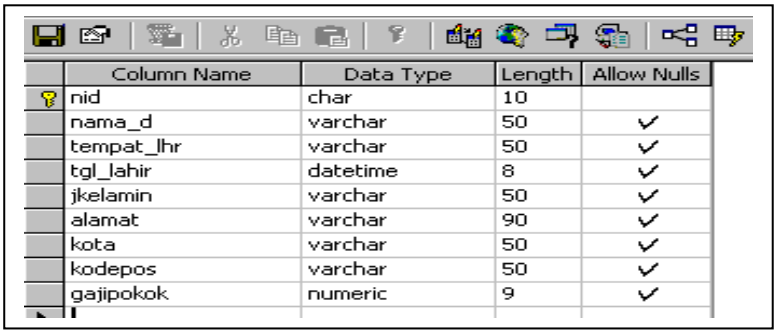
3.2.1. *Data Definition Language* (DDL)

Struktur / skema basis data yang menggambarkan / mewakili desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yaitu DDL. Dengan bahasa ini kita dapat membuat tabel (*create table*) baru, *indeks*, mengubah *table*, menentukan struktur penyimpanan *table*, dan lainnya. Hasil dari kompilasi perintah DDL, adalah kumpulan table yang disimpan dalam *file* khusus yang disebut kamus data (*data dictionary*).

Kamus data merupakan suatu metadata (superdata), yaitu data yang mendiskripsikan data sesungguhnya. Contoh perintah DDL dengan Foxpro adalah *create* matakuliah, *modify report*, *modify structure*, dan lainnya, sedangkan perintah DDL dengan MS-SQl Server 2000, contohnya adalah *create new database* Penjadwalan_mengajar_dosen dan contoh lainnya adalah *new table* dosen dengan struktur sebagai berikut:



Gambar 3.1. Create database dengan MS-SQL Server 2000



Column Name	Data Type	Length	Allow Nulls
nid	char	10	
nama_d	varchar	50	✓
tempat_lhr	varchar	50	✓
tgl_lahir	datetime	8	✓
jkelamin	varchar	50	✓
alamat	varchar	90	✓
kota	varchar	50	✓
kodepos	varchar	50	✓
gajipokok	numeric	9	✓

Gambar 3.2. Create table mahasiswa dengan MS-SQL Server 2000

3.2.2. Data Manipulation Language (DML)

Bentuk bahasa basis data untuk melakukan menipulasi dan pengambilan data pada suatu basis data. Manipulasi data pada dabase dapat berupa :

1). Penyisipan / penambahan data pada *file / table* dalam suatu basis data.
2). Penghapusan data pada *file / table* dalam suatu basis data.
3). Pengubahan data pada *file / table* dalam suatu basis data.
4). Penelusuran data pada *file / table* dalam suatu basis data.

Pada level phisik kita harus mendefinisikan alghoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi yang dipentingkan bukan efisien akses, tapi juga efisiensi interaksi pemakai dengan sistem.

DML merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 (dua) jenis DML adalah

sebagai berikuit :

1). **Prosedural**, yang mensyaratkan pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.

2). **Nonprosedural**, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

Contoh paket bahasa prosedural DML : dBase, FoxBase, sedang untuk Nonprosedural DML : SQL (Structure Query Language), QBE (Query By Example).

Query

Adalah pernyataan yang diajukan untuk mengambil informasi di dalam suatu basis data. Query merupakan bagian dari DML yang untuk pengambilan informasi disebut Query Language.

Berikut ini adalah contoh penggunakan perintah query dengan menggunakan database MS-SQL Server 2000, dapat dilihat pada gambar 3.3. berikut ini :

Perintah Query (SQL)

```
SELECT nid,nama_d,jkelamin,alamat,kota
FROM Dosen
Where jkelamin='Pria'
```

Hasil Query

	nid	nama_d	jkelamin	alamat	kota
1	00001	Riswoko Sasono,MMSi	Pria	Jl. Catur 3 No.23	Bekasi Utara
2	01001	Cokro Diningrat,Ir,MT	Pria	Jl. Dayang Raya 2 No.12	Cibitung
3	01002	Sakib Aljaber,MT	Pria	Jl. Kemuning 1 No.1	Cikarang
4	95001	Bambang Sutedjo,Ir,MMSi	Pria	Jl. Mawar 6 No.2	Jakarta Selatan
5	96001	Triyatno,Ir,MM,M.Kom	Pria	Jl. Mawar Indah 1 No.1	Bekasi Barat
6	97002	Djoko Pamungkas,M.Kom	Pria	Jl. Anggrek 7 No.1	Bekasi Timur
7	98001	Didik Atmadja,Ir,MMSi	Pria	Jl. Cipete Raya No.6	Jakarta Selatan
8	98002	Bagus Windarjo,M.Kom.	Pria	Jl. H. Ali No.21	Cibitung

Gambar 3.3. Perintah Query dengan MS-SQL Server 2000

3.3. Pengguna Database

Basis data yang telah diciptakan yang terdiri dari beberapa table yang saling berkaitan, tentunya harus dimanfaatkan oleh para pengguna. Di dalam suatu basis data terdapat beberapa pengguna, antara lain adalah:

1). ***Database* Manager**

Satu *database* manager adalah satu modul program yang menyediakan *interface* antara penyimpanan data *low-level* dalam *database* dengan satu aplikasi program dan *query* yang diajukan ke sistem.

Tugas dan tanggungjawab *database manager*:

a. Interaksi dengan manager file
b. Integritas
c. Keamanan
d. *Bakcup* dan *recovery*

2). ***Database Administrator***

Orang yang mempunyai kekuasaan sebagai pusat pengontrolan terhadap seluruh sistem baik data maupun program yang mengakses data disebut sebagai *database administrator*.

Fungsi database administrator:

a. Mendefinisikan pola struktur database.
b. Mendefinisikan struktur penyimpanan dan metode akses.
c. Mampu memodifikasi pola dan organisasi phisik.
d. Memberikan kekuasaan pada user untuk mengakses data.
e. Menspesifikasikan keharusan integritas data.

3). **Database User**

Ada 4 macam pemakai database yang berbeda keperluan dan cara aksesnya, yaitu : 1. Programmer Aplikasi, 2. Casual User (user mahir), 3. User Umum (end user), dan 4. User khusus (specialized user).

3.4. Basis Data Relasional

Pada model relasional, basis data akan “disebar” (dipilah-pilah) kedalam berbagai tabel 2 dimensi. Setiap tabel terdiri atas lajur mendatar yang disebut dengan baris data

(row / record), dan lajur vertical yang biasa disebut dengan kolom (columm / field). Di setiap pertemuan baris data dan kolom itulah item-item data (satuan data terkecil) ditempatkan.

Ilustrasi Model Data Relasional

Sebelum penerapan basis data dilakukan, lebih dahulu kita harus mengetahui contoh data yang akan digunakan. Contoh data ini harus dianalisis untuk mengetahui karakteristik mengenai masing-masing tabel data. Pengetahuan karakteristik dijadikan dasar untuk penetapan struktur dari masing-masing tabel. Didalam penerapan basis data, penetapan *struktur table* dilakukan sebelum penyimpanan data.

Berikut contoh data yang akan digunakan (data yang ada tidak menggambarkan kenyataan yang sesungguhnya).

1. Data dosen, dengan isi data sebagai berikut:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
95001	Bambang Sutedjo,Ir,MMSi	Jakarta	2/23/58	Pria	Jl. Mawar 6 No.2	Jakarta Selatan	23234	1300000
95002	Asri Kasetyaningsih,M.Kom	Semarang	12/25/62	Wanita	Jl. Perjuangan 3 No.11	Bekasi Timur	54567	1200000
96001	Triyatno,Ir,MM,M.Kom	Bekasi	5/14/67	Pria	Jl. Mawar Indah 1 No.1	Bekasi Barat	54356	1100000
96002	Diastuti Pujiningsih,MM,MT	Surabaya	12/24/70	Wanita	Jl. Janur Kuning 2 No.56	Cibitung	54566	1000000
97001	Endang Junianti,Ir,MMSi	Bandung	5/20/67	Wanita	Jl. Kemukus 2 No.56	Cikarang	56555	1150000
97002	Djoko Pamungkas,M.Kom	Bogor	5/28/71	Pria	Jl. Anggrek 7 No.1	Bekasi Timur	45666	1100000
98001	Didik Atmadja,Ir,MMSi	Bandung	4/20/70	Pria	Jl. Cipete Raya No.6	Jakarta Selatan	24123	1250000
98002	Bagus Windarjo,M.Kom.	Tangerang	12/13/71	Pria	Jl. H. Ali No.21	Cibitung	54523	1150000
99001	Dewi Anjani,Ir,MM	Bekasi	12/14/75	Wanita	Jl.Kemang Raya 2 No.1	Bekasi Barat	54567	1200000
00001	Riswoko Sasono,MMSi	Bogor	12/24/74	Pria	Jl. Catur 3 No.23	Bekasi Utara	52123	1300000
00002	Hasta Riyanti,Ir,MM	Jakarta	5/27/69	Wanita	Jl. Delima 2 No.2	Bekasi Timur	45612	1150000
01001	Cokro Diningrat,Ir,MT	Bekasi	12/19/66	Pria	Jl. Dayang Raya 2 No.12	Cibitung	54456	1200000
01002	Sakib Aljaber,MT	Cikarang	5/20/70	Pria	Jl. Kemuning 1 No.1	Cikarang	56234	1150000

Karakreristik data :

- Memiliki 9 buah kolom data.
- Kolom pertama berisi data angka tetapi tidak menunjukkan suatu jumlah (disebut alfa numerik) dengan lebar (banyaknya karakter) tetap, 5 karakter / dijit.
- Kolom kedua berisi data karakter / string dengan lebar maksimum 30 karakter.
- Kolom ketiga berisi data karakter / string dengan lebar maksimum 10 karakter.
- Kolom keempat berisi data penanggalan / *date.*
- Kolom kelima berisi data karakter / string dengan lebar maksimum 6 karakter.
- Kolom keenam berisi data karakter / string dengan lebar maksimum 50 karakter.
- Kolom ketujuh berisi data karakter / string dengan lebar maksimum 20 karakter.
- Kolom kedelapan berisi data karakter / string dengan lebar maksimum 5 karakter.

- Kolom kesembilan berisi data numeric dengan lebar maksimum 8 karakter.

Contoh diatas adalah karakteristik data untuk data dosen, data ini diambilkan dari contoh database pada sub bab 2.5 (contah database). Atas dasar karakteristik data tersebut selanjutnya kita bisa membentuk database physical (dengan catatan table – table yang telah terbentuk telah dilakukan normalisasi data) dengan menggunanan MS-Access, SQL Server, MYSql, Oracle atau *tool database* lainnya.

Berangkat dari mendefinisikan karakteristik tersebut di atas, maka kita dapat menetapkan struktur dari masing-masing *table*. Secara minimal yang harus ditentukan dalam struktur *table* adalah :

1). Nama kolom (*field / atribut*).
2). Tipe data (*data type*)
3). Lebar (banyaknya karakter / dijit maksimum yang dapat ditampung.
4). Pendefinisian kolom (apakah *null* atau *not null*)

Berangkat dari karakteristik yang dapat kita simpulkan dari fakta yang ada dari masing-masing data, struktur dari *table* dosen pada basis data Penjadwalan_mengajar_dosen, yang dapat ditentukan adalah sebagai berikut (struktur dengan MS-SQL Server):

1. *Table* dosen

	Column Name	Data Type	Length	Allow Nulls
🔑	nid	char	10	
	nama_d	varchar	50	✓
	tempat_lhr	varchar	50	✓
	tgl_lahir	datetime	8	✓
	jkelamin	varchar	50	✓
	alamat	varchar	90	✓
	kota	varchar	50	✓
	kodepos	varchar	50	✓
	gajipokok	numeric	9	✓

Adapun untuk struktur untuk *table* Matakuliah Jurusan dan Mengajar adalah sebagai berikut:

2. *Table* matakuliah

	Column Name	Data Type	Length	Allow Nulls
🔑	kdmk	char	9	
	nama_mk	char	50	✓
	sks	int	4	✓
	semester	char	2	✓

3. Tabel jurusan

	Column Name	Data Type	Length	Allow Nulls
🔑	kode_jur	char	2	
	nama_jur	varchar	30	
	jenjang	varchar	10	✓
	nama_kajur	varchar	50	✓

4. Tabel mengajar

	Column Name	Data Type	Length	Allow Nulls
🔑	nid	char	10	
🔑	thn_akademik	char	4	
🔑	smt	int	4	
🔑	hari	varchar	10	
🔑	jam_ke	char	1	
	kdmk	char	9	✓
	waktu	char	10	✓
	kelas	char	5	✓
	kode_jur	char	2	✓

Untuk kolom *Allow Nulls* jika tidak ada tanda *check list* maka kolom (*field*) tersebut harus isi datanya dan tidak boleh kosong.

Untuk tanda *key* adalah kolom (*field*) tersebut adalah sebagai *primary key* pada table yang terbentuk. Pendefinisan untuk tipe data dan lebar *field* harus disesuaikan dengan data sesungguhnya, lebar tidak boleh kurang tapinhindari lebar yang berlebihan.

Pertanyaan Soal

1. Jelaskan dan berikan contoh pengertian dari DDL (*data definition language*) ?.
2. Jelaskan pengertian dari DML (*data manipulation language*), manipulasi apa saja yang dapat kita lakukan berkenaan dengan basis data (*database*) ?.
3. Jelaskan pengertian daripada *Query* ?.

4. Jelaskan pengertian *database administrator*, fungsi apa saja yang harus dilakukan oleh *database administrator* ?.
5. Berikan penjelasan tentang karakteristik data untuk *table* matakuliah, jurusan dan mengajar ?.