BAB 2
OPERASI DASARI BASIS DATA

2.1. Operasi Dasar

Didalam sebuah disk (*hard disk*), basis data dapat diciptakan dan dapat pula ditiadakan. Dalam sebuah disk kita dapat menempatkan beberapa (lebih dari satu) basis data (Misalnya basis data nilaiMahasiswa, kepegawaian, keuangan, penjualan, pepustakaan dan lainnya). Sementara dalam sebuah basis data kita dapat menempatkan satu atau lebih *file / table*. Misalkan dalam basis data penjualan terdiri dari table barang, faktur, pelanggan dan transaksi barang.

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data adalah sebagai berikut:

1). *Pembuatan basis data baru (create database*), identik dengan pembuatan lemari arsip yang baru.
2). *Penghapusan basis data (drop database)*, identik dengan perusakan lemari arsip, sekaligus beserta isinya jika ada.
3). *Pembuatan table baru ke suatu basis data (create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4). *Penghapusan table dari suatu basis data (drop table),* identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5). *Penambahan / pengisian data baru di sebuah basis data (insert),* identik dengan penambahan lembaran arsip ke sebuah map arsip.
6). *Pengambilan data dari sebuah table (retrieve / search),* identik dengan pencarian lembaran arsip dalam sebuah map arsip.
7). *Pengubahan data dalam sebuah table (update)*, identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
8). *Penghapusan data dari sebuah table (delete)*, identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

Operasi pembuatan basis data dan tabel merupakan operasi awal yang hanya

dilakukan sekali dan berlaku seterusnya. Sedangkan untuk operasi pengisian, perubahan, penghapusan dan pencarian data merupakan operasi rutin yang berlaku berulang-ulang.

2.2. Kegunaan Database

Penyusunan satu basis data digunakan untuk mengatasi masalah – masalah pada penyusunan data, antara lain:

1). Redundansi dan inkonsistensi data
2). Kesulitan pengaksesan data
3). Isolasi data untuk standarisasi
4). *Multiple user* (banyak pemakai)
5. Masalah keamanan (*security*)
6. Masalah integrasi (*kesatuan*)
7. Masalah *data independence* (kebebasan data)

2.2.1. Redundansi dan inkonsistensi data

Jika *table* dan program aplikasi diciptakan oleh *programmer* yang berbeda dengan waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada *table* yang berbeda pada suatu *database.*

C o n t o h

Nama, alamat, dan telpon dari mahasiswa di sebuah Perguruan Tinggi tercatat pada table Anggota (pada database Perpustakaan), KRS dan juga pada Keuangan. Apabila kita berbicara sistem yang berbasis jaringan maka 1(satu) *table* mahasiswa bisa dimanfaatkan oleh beberapa sub database yang menginginkannya.

Penyimpanan dibeberapa tempat untuk data yang sama ini disebut sebagai *redundansi* dan mengakibatkan pemborosan ruang penyimpanan dan juga biaya untuk akses lebih tinggi.

Penyimpanan data yang sama berulang-ulang di beberapa tempat dalam *database* dapat mengakibatkan juga *inkonsistensi* (tidak konsisten). Hal ini terjadi, bila suatu ketika mahasiswa tersebut pindah alamat berubah, maka seluruh *table*

yang memuat data tersebut harus diubah / *update.* Bila salah satu saja yang *diupdate*, maka menjadi tidak konsisten.

2.2.2. Kesulitan dalam pengaksesan data

Pada suatu saat dibutuhkan untuk mencetak data siapa saja mahasiswa yang berada di kota 'Bekasi Timur', padahal belum tersedia program yang telah ditulis untuk mengeluarkan data tersebut. Maka kesulitan akan timbul, dan penyelesaian ke arah itu adalah DBMS yang mampu mengambil data secara langsung dengan bahasa yang familiar dan mudah digunakan (*user frindly*).

2.2.3. Isolasi data untuk standarisasi

Jika data tersebar dalam beberapa *file / table* dalam bentuk format yang tidak sama, maka ini akan menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka haruslah data dalam satu basis data dibuat satu format, sehingga mudah dibuat program aplikasinya.

2.2.4. *Multiple User* (Banyak Pemakai)

Dalam rangka mempercepat semua daya guna sistem dan mendapat responsi waktu yang cepat, beberapa sistem mengijinkan banyak pemakai untuk meng *"update "* data secara simultan. Salah satu alasan mengapa basis data dibangun karena nantinya data tersebut digunakan oleh banyak orang dalam waktu yang sama atau berbeda, diakses oleh program yang sama tapi berbeda orang dan waktu.

Semua itu memungkinkan terjadi, karena data yang diolah tidaklah tergantung dan menyatu dalam program tapi ia terlepas dalam satu kelompok data.

2.2.5. Masalah keamanan (*security*)

Tidak semua pemakai sistem basis data diperbolehkan untuk mengakses semua data. Misalkan data mengenai gaji seorang karyawa hanya boleh dibuka oleh bagian keuangan dan personalia, tidak diperkenankan bagian gudang membaca dan mengubahnya.

2.2.6. Masalah integritas (kesatuan)

Basis data berisi *file / table* yang saling terkait, masalah utama adalah

bagaimana kaitan antar *table* itu terjadi. Meskipun kita mengetahui *table* A berkaitan dengan *table* B, namun secara teknis ada *field / atribut* kunci yang mengaitkan / merelasikan *table* tersebut.

2.2.7. Masalah *data independence* (kebebasan data)

Paket bahasa yang diciptakan oleh DBMS, perubahan pada struktur *file / table*, setiap kali kita hendak melihat data cukup dengan *utility list*, menambah data dengan *Append* (misal untuk DBMS Clipper atau Foxpro), merubah struktur table dengan *Design Table*, melakukan penelurusan data dengan *query* (misal untuk Access, Sql Server, MySql atau Oracle). Ini berarti perintah-perintah dalam paket DBMS bebas terhadap basis data. Apapun perubahan dalam basis data semua perintah akan mengalami kestabilan tanpa mengalami perubahan.

2.3. Keuntungan Sistem Basis Data

Keuntungan-keuntungan yang dapat diperoleh dari penerapan sistem basis data pada suatu perusahaan adalah sebagai berikut:

1). *Mengurangi redudansi data*
Data yang sama pada beberapa aplikasi cukup disimpan sekali saja.

2). *Integritas Data*
Dimana data terismpan secara akurat karena tidak adanya redudansi data.

3). *Menghindari inkonsisten data*
Sebagai akibat tidak adanya data yang redundansi data, sehingga tidak terjadi inkonsisten data, karena data yang akan diupdate cukup dilakukan sekaligus saja.

4). *Penggunaan data bersama*
Data yang sama dapat diakses atau dimanfaatkan oleh beberapa user pada saat yang bersamaan.

5). *Standarisasi data*
Akibat tidak adanya redundansi, inkonsisten, dan integritas data, maka akan terciptanya adanya standarisasi data.

6). *Jaminan Keamanan Data (Security Data)*
Data yang tersimpan hanya dapat diakses oleh yang mempunyai otoritas terhadap data tersebut.

7). *Menyeimbangkan kebutuhan data*

Data ditentukan prioritas suatu operasi, misalkan antara update dengan retrieve data.

2.4. Kerugian Sistem Basis Data

Kerugian - kerugian yang ada dengan diterapkannya basis data pada suatu perusahaan adalah sebagai berikut:

1). Diperlukan *harware* (perangkat keras tambahan) : CPU yang lebih kuat, terminal yang lebih banyak, alat komunikasi.
2). Biaya *Performance* yang lebih besar : listrik, personil yang lebih tinggi klasifikasinya, biaya telekomunikasi antar lokasi.
3). Rawannya keberhasilan operasi : gangguan listrik, dan komunikasi.
4). Sistem kelihatan lebih kompleks : banyaknya aspek yang harus diperhatikan.

2..5. Contoh *Database*

Berikut ini ini adalah contoh dari *database* Penjadwalan_mengajar_dosen pada STMIK 'Revanda Jaya' Bekasi, dimana pada database tersebut terdiri dari *file / table* Dosen, Matakuliah, Jurusan dan Mengajar.

1. Dosen

Dosen ={**nid**,nama_d,tempat_lhr,tgl_lahir,jkelamin,alamat,kota,kodepos,gajipokok}

Dengan *data value* 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

Gambar 2.1. *Data value* Dosen

2. Matakuliah

   Matakuliah ={**kdmk**,nama_mk,sks,semester}

   Dengan *data value* sebagai berikut:



kdmk	nama_mk	sks	semester
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1
MKB331203	PRAKTIKUM PAKET PROGRAM APLIKASI I	1	1
MKK231201	DASAR MANAJEMEN DAN BISNIS I	2	1
MKK231203	MATEMATIKA DISKRIT I	2	1
MKK231205	AKUTANSI DASAR I	2	1
MKK231213	PENGANTAR EKONOMI	2	1
MPK131201	PENDIDIKAN AGAMA I	2	1
MPK131203	PENDIDIKAN PANCASILA	2	1
MPK131205	BAHASA INGGRIS I	2	1
MPK231207	SISTEM BASIS DATA	2	1
MPK131206	BAHASA INGGRIS II	2	2
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2
MKK231206	AKUTANSI DASAR II	2	2
MKK231204	MATEMATIKA DISKRIT II	2	2
MKK231202	DASAR MANAJEMEN DAN BISNIS II	2	2
MKB331209	ALGORITMA PEMROGRAMAN I (PASCAL)	3	2
MKB331210	PRAK. ALGORITMA PEMROGRAMAN I (PASCAL)	1	2
MKB331205	PERANCANGAN BASIS DATA	4	2
MKB331206	PAKET PROGRAM APLIKASI II	1	2

Gambar 2.2. *Data value* Matakuliah

3. Jurusan

   Jurusan ={**kode_jur**,nama_jur,sjenjang,nama_kajur}

   Dengan *data value* sebagai berikut:



kode_jur	nama_jur	jenjang	nama_kajur
KA	Komputerisasi Akuntansi	Diploma 3	Rini Wulandari,MM,MMSi
MI	Manajemen Informatika	Diploma 3	Wahono Diprodjo, MM, MKom.
SI	Sistem Informasi	Strata 1	Bagus Hermansyah, MM.Si,M.Kom.
TI	Teknik Informatika	Strata 1	Fadjar Sasongko, MT, M.Kom.
TK	Teknik Komputer	Diploma 3	Agus Budiyantara, T, MT.

Gambar 2.3. *Data value* Jurusan

4. Mengajar

   Mengajar ={**nid,thn_akademik,smt,hari,jam_ke**,kdmk,waktu,kelas,kode_jur}

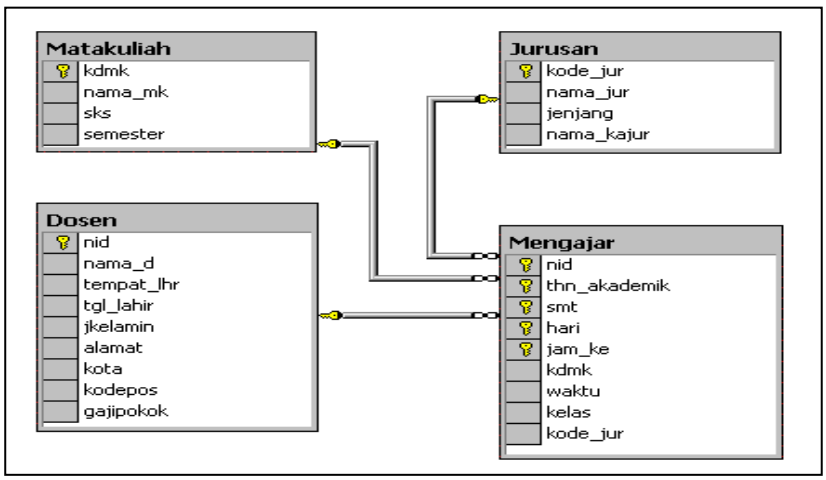
   Dengan *data value* sebagai berikut:



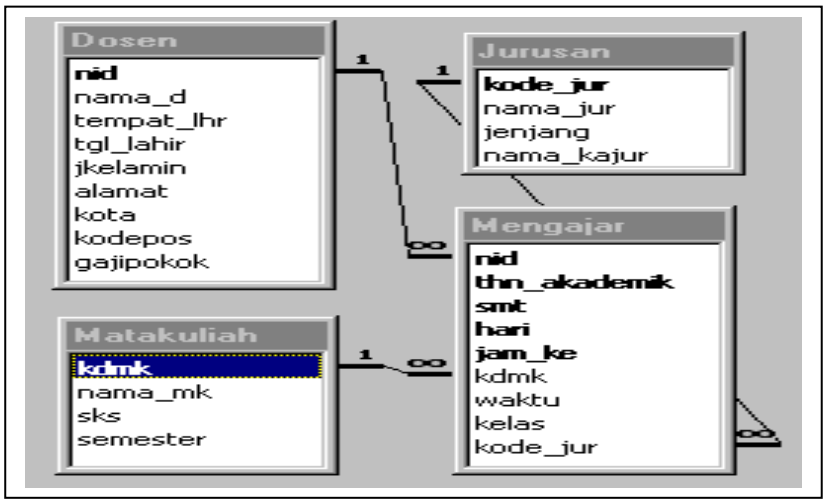
nid	thn_akademik	smt	hari	jam_ke	kdmk	waktu	kelas	kode_jur
00001	2004	1	Rabu	1	MKB331205	8:00	T202	TI
00001	2004	1	Senin	1	MKB331201	8:00	M101	MI
00002	2004	2	Jumat	1	MPK131204	8:00	S201	SI
00002	2004	2	Jumat	2	MKP131204	10:00	S202	SI
95001	2004	1	Kamis	1	MKB331201	8:00	T101	TI
95001	2004	1	Senin	1	MKB331201	8:00	M101	MI
98002	2004	2	Rabu	1	MKB331204	8:00	S201	SI
98002	2004	2	Selasa	2	MPK131204	10:00	M201	MI
99001	2004	1	Senin	1	MKB331201	8:00	M102	MI
99001	2004	2	Selasa	1	MKB331205	8:00	T201	TI

Gambar 2.4. *Data value* Mengajar

Implementasi relasi (hubungan antar *table*) yang ada pada *database* tersebut dapat dilihat pada gambar berikut ini (Dengan database MS-SQL Server 2000 dan MS-Access):



Gambar 2.5. Diagram database Penjadwalan_mengajar_doseen dengan SQL-Server 2000



Gambar 2.6. Relationship database Penjadwalan_mengajar_doseen dengan MS-Access

Pada gambar tersebut diatas terlihat bahwa pada *table* mengajar, berelasi kepada *table* dosen, matakuliah dan jurusan, dikarenakan pada *table* mengajar tersebut membutuhkan data – data yang ada pada ketiga tabel tersebut, artinya:

- Seorang dosen bisa mengajar lebih dari satu matakuliah pada semester yang sama.
- Satu matakuliah bisa diajar (diampu) oleh banyak dosen dan jurusan.

Operasi manipulasi yang dapat dilakukan pada *database* tersebut adalah sebagai berikut:

1). *Insert* : Kita dapat melakukan pemasukan data – data baru pada *file / table* dosen, matakuliah, jurusan dan mengajar.

2). *Delete* : Kita dapat melakukan penghapusan data yang telah ada pada *file / table* dosen, matakuliah, jurusan dan mengajar untuk data – data yang tidak diperlukan lagi.

3). *Update* : Kita dapat melakukan perubahan data – data alamat dan kota seorang dosen pada *file / table* dosen dikarenakan dosen tersebut pindah alamat, dan kita dapat melakukan perubahan untuk data yang lainnya pada *database* tersebut.

4). *Retrieve* : Kita dapat menampilkan Informasi mengenai dosen menurut jenis kelamin, kota alamat dan lainnya, informasi mengenai transaksi mengajar dosen, informasi jurusan, informasi mengenai matakuliah berdasarkan sks nya dan informasi lainnya.

2.6. Abstraksi Data

Kegunaan utama sistem basis data adalah agar pemakai (*user)* mampu menyusun suatu pandangan abstaksi dari data. Pemakai atau *user* dapat dikelompokkan menjadi tiga tingkatan abstaksi saat memandang suatu basis data, yaitu :

1). Level *Phisik.*

2). Level *Konseptual.*

3). Level Padangan Pemakai (*View*)

2.6.1. Level Phisik

Level abstraksi paling rendah, menggambarkan bagaimana (*how*) data disimpan dalam kondisi sebenarnya. Level ini tentu paling kompleks, struktur data level terendah digambarkan pada level ini. Level ini digunakan oleh *programmer*, yang digunakan untuk melakukan pemrograman dengan mengunakan database dan DBMS tertentu sesuai dengan kebutuhan daripada *end-user.*

2.6.2. Level Konseptual

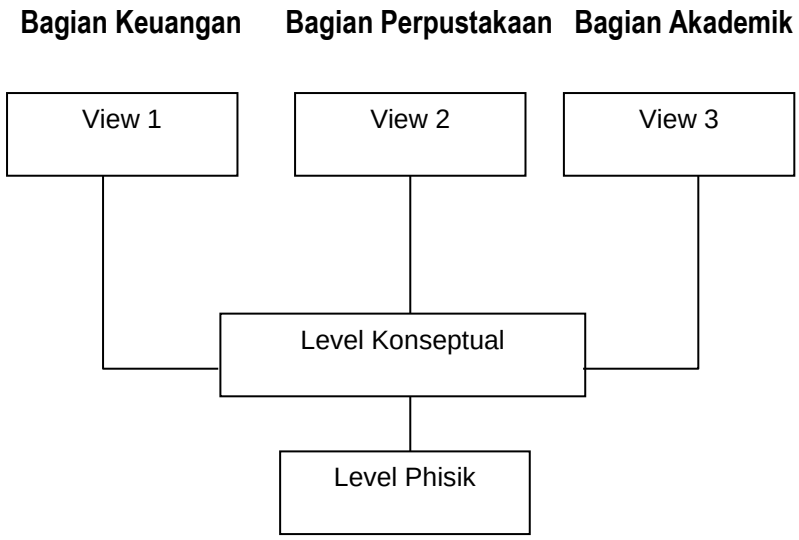
Level abstraksi data level lebih tinggi yang menggambarkan data apa (*what*) yang disimpan dalam basis data, dan hubungan relasi yang terjadi antar data. Level ini menggambarkan keseluruhan basis data. Pemakai tidak memperdulikan kerumitan dalam struktur level phisik lagi, penggambaran cukup dengan memakai kotak, garis dan keterangan secukupnya. Level ini digunakan oleh *database administrator*, yang memutuskan informasi apa yang akan dipelihara dalam satu *database.*

2.6.3. Level Pandangan Pemakai (View Level)

Level abstraksi tertinggi yang mengambarkan hanya satu bagian dari keseluruhan database. Bila pada level konseptual data merupakan suatu kumpulan besar dan kompleks, pada level ini hanya sebagian saja yang dilihat dan dipakai. Hal ini disebabkan beberapa pemakai database tidak membutuhkan semua isi database. Level ini sangat dekat dengan pemakai (*user*), dan setiap user kemungkinan hanya membutuhkan sebagian dari database. Ada beberapa kelompok user dengan pandangan berbeda butuh data dalam *database*, jadi pada level ini yang memakai adalah pemakai akhir atau *end-user.*

Misalkan pemakai akhir pada bagian keuangan hanya memakai data untuk *file / table* pembayaran, mahasiswa dan karyawan, tetapi tidak membutuhkan *file / table* buku dan nilai. Demi kemudahan interaksi antara pemakai dengan sistem, maka view level ini didefinisikan. Jadi ada beberapa pandangan disusun untuk mengakses satu sistem *database* yang sama.

Hubungan antar level tersebut dapat dilihat pada gambar 2.7 berikut ini :



Gambar 2.7. Abstraksi Data

Pertanyaan Soal

1. Sebutkan dan jelaskan operasi – operasi dasar yang berkenaan dengan basis data (*database*) ?.
2. Jelaskan dan berikan contoh masalah *redundancy* dalam suatu basis data (*database*) ?.
3. Jelaskan keuntungan – keuntungan apa saja dengan diterapkannya basis data (*database*) pada suatu perusahaan ?.
4. Jelaskan kerugian - kerugian apa saja dengan diterapkannya basis data (*database*) pada suatu perusahaan ?.
5. Berikan contoh suatu *database*, kelompokkan mana sebagai *file / table*, dan *field / atribut*nya ?
6. Jelaskan operasi manipulasi apa saja yang dapat dilakukan berkenaan dengan pembentukan basis data (*database*) ?.
7. Jelaskan pengertian level konseptual pada abstraksi data ?.