

## BASIS DATA (DATABASE)

### **Data, Informasi dan Basis Data**

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*) [Ramez2000]. Sedangkan menurut George Tsu-der Chou basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang [Abdul1999]. Menurut *Encyclopedia of Computer Science and Engineer*, para ilmuwan di bidang informasi menerima definisi standar informasi yaitu data yang digunakan dalam pengambilan keputusan.

Definisi lain dari basis data menurut Fabbri dan Schwab adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan duplikasi data.

Menurut Ramez Elmasri mendefinisikan basis data lebih dibatasi pada arti implisit yang khusus, yaitu:

- a. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real world*).
- b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
- c. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa *user* dan beberapa aplikasi yang sesuai dengan kepentingan *user*.

Dari beberapa definisi-definisi tersebut, dapat dikatakan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa user untuk berbagai kepentingan [Waliyanto2000].

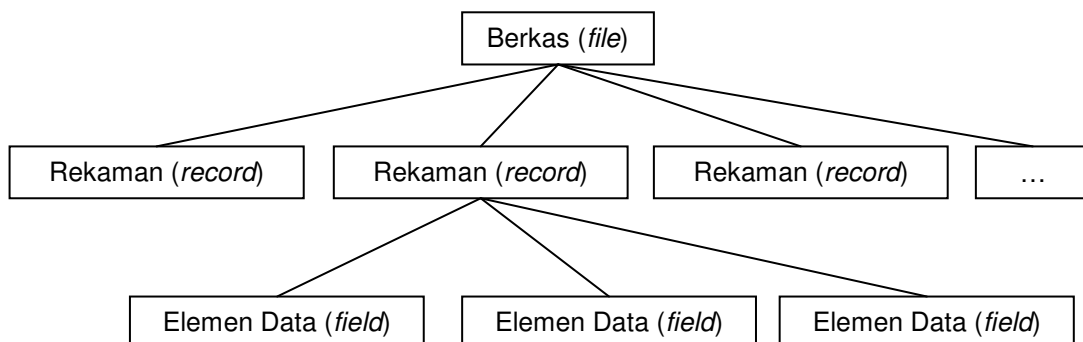
### **Hirarki Data**

Data diorganisasikan kedalam bentuk elemen data (*field*), rekaman (*record*), dan berkas (*file*). Definisi dari ketiganya adalah sebagai berikut:

Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama, Alamat, Telepon atau Jenis Kelamin.

Rekaman merupakan gabungan sejumlah elemen data yang saling terkait. Istilah lain dari rekaman adalah baris atau tupel.

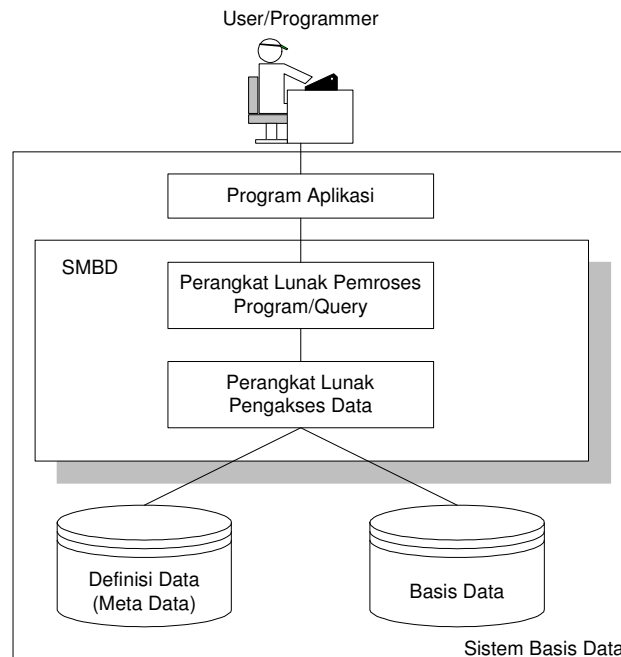
Berkas adalah himpunan seluruh rekaman yang bertipe sama.



**Gambar 1.1 Hirarki data**

## Sistem Basis Data

[Waliyanto2000] Gabungan antara basis data dan perangkat lunak SMDB (Sistem Manajemen Basis Data) termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem disebut dengan Sistem Basis Data.



**Gambar 1.2 Konsep Sistem Basis Data (kompilasi Ramez Elmasri. dkk 1994)**

C. J. Date menyatakan bahwa sistem basis data dapat dianggap sebagai tempat untuk sekumpulan berkas data yang terkomputerisasi dengan tujuan untuk memelihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan.

## ***Data Base Management System (DBMS)/Sistem Manajemen Basis Data (SMB)***

DBMS dapat diartikan sebagai program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memodifikasi dan memperoleh data/informasi dengan praktis dan efisien.

Kelebihan dari DBMS antara lain adalah:

- **Kepraktisan.** DBMS menyediakan media penyimpan permanen yang berukuran kecil namun banyak menyimpan data jika dibandingkan dengan menggunakan kertas.
- **Kecepatan.** Komputer dapat mencari dan menampilkan informasi yang dibutuhkan dengan cepat.
- **Mengurangi kejemuhan.** Pekerjaan yang berulang-ulang dapat menimbulkan kebosanan bagi manusia, sedangkan mesin tidak merasakannya.
- **Update to date.** Informasi yang tersedia selalu berubah dan akurat setiap.

[Waliyanto2000] Keuntungan-keuntungan dalam penggunaan DBMS antara lain adalah:

- a. Pemusatan kontrol data. Dengan satu DBMS di bawah kontrol satu orang atau kelompok dapat menjamin terpeliharanya standar kualitas data dan keamanan batas penggunaannya serta dapat menetralkan konflik yang terjadi dalam persyaratan data dan integritas data dapat terjaga.
- b. Pemakaian data bersama (Shared Data). Informasi yang ada dalam basis data dapat digunakan lebih efektif dengan pemakaian beberapa user dengan kontrol data yang terjaga.
- c. Data yang bebas (independent). Program aplikasi terpisah dengan data yang disimpan dalam komputer.
- d. Kemudahan dalam pembuatan program aplikasi baru.
- e. Pemakaian secara langsung. DBMS menyediakan interface yang memudahkan pengguna dalam mengolah data.

- f. Data yang berlebihan dapat dikontrol. Data yang dimasukkan dapat terjadi kerangkapan (*redundant*), untuk itu DBMS berfungsi untuk menurunkan tingkat redundancy dan pengelolaan proses pembaruan data.
- g. Pandangan user (*user view*). Ada kemungkinan basis data yang diakses adalah sama, maka DBMS mampu mengatur interface yang berbeda dan disesuaikan dengan pemahaman tiap user terhadap basis data menurut kebutuhan.

Kelemahan-kelemahan DBMS antara lain:

- a. Biaya. Kebutuhan untuk mendapatkan perangkat lunak dan perangkat keras yang tepat cukup mahal, termasuk biaya pemeliharaan dan sumber daya manusia yang mengelola basis data tersebut.
- b. Sangat kompleks. Sistem basis data lebih kompleks dibandingkan dengan proses berkas, sehingga dapat mudah terjadinya kesalahan dan semakin sulit dalam pemeliharaan data.
- c. Resiko data yang terpusat. Data yang terpusat dalam satu lokasi dapat beresiko kehilangan data selama proses aplikasi.

### Model Data

Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu:

- a. Model data konseptual (*high level*) menyajikan konsep tentang bagaimana user memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyajian data yaitu:
  - **Entity** (entitas) merupakan penyajian obyek, kejadian atau konsep dunia nyata yang keberadaannya secara eksplisit didefinisikan dan disimpan dalam basis data, contohnya Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.
  - **Attribute** (atribut) adalah keterangan-keterangan yang menjelaskan karakteristik dari suatu entitas seperti NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
  - **Relationship** (hubungan) merupakan hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibelinya.
- b. Model data fiskal (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pemcarian rekaman data lebih efisien.
- c. Model data implementasi (*representational*) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para user mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan oleh model hirarki, jaringan dan relasional.

### Skema dan Instan Basis Data

Skema basis data merupakan deskripsi dari basis data yang spesifikasinya ditentukan dalam tahap perancangan namun tidak terlalu diharapkan diubah setiap saat. Penggambaran skema umumnya hanya berisi sebagian dari detail deskripsi basis data.

MAHASISWA

NIM	NAMA	FAKULTAS	JURUSAN
-----	------	----------	---------

MATA KULIAH

KD_MK	MATA KULIAH	SKS
-------	-------------	-----

DOSEN

KD_DOSEN	NAMA	ALAMAT	TELEPON
----------	------	--------	---------

KULIAH

ID_KUL	NIM	KD_MK	KD_DOSEN	SEMESTER	TAHUN	NILAI
--------	-----	-------	----------	----------	-------	-------

**Gambar 1.3 Skema Basis Data Akademi**

Sekelompok data yang tersusun dalam satu baris rekaman (*record/tuple*) dan tersimpan dalam basis data disebut dengan instansi (*instance*) atau kejadian (*occurences*).

## Arsitektur DBMS

Arsitektur ini dikenal dengan nama arsitektur tiga skema (*three-schema architecture*) dimana fungsi ini untuk memisahkan antara basis data fisik dengan program aplikasi user. Skema-skema tersebut adalah sebagai berikut:

- a. Level internal merupakan skema internal yang memuat deskripsi struktur penyimpanan basis data dan menggunakan model data fisik serta mendefinisikan secara detail penyimpanan data dalam basis data, serta jalur pengaksesan data.
- b. Level konseptual adalah skema yang memuat deskripsi struktur basis data secara keseluruhan untuk semua pemakai. Skema ini hanya memuat deskripsi tentang entitas, atribut, hubungan dan batasan, tanpa memuat deskripsi data secara detail.
- c. Level eksternal merupakan skema eksternal (*user view*) yang mendefinisikan pandangan data terhadap sekelompok user (*local view*) dengan menyembunyikan data lain yang tidak diperlukan oleh kelompok user tersebut.

Keuntungan dari arsitektur ini antara lain:

- a. Perubahan skema konseptual, yaitu adanya perubahan dalam skema konseptual contohnya penambahan suatu item data tidak akan berpengaruh pada program aplikasi. Tetapi jika skema eksternal tidak sesuai lagi dengan skema konseptual yang baru maka program aplikasi harus disesuaikan juga.
- b. Perubahan skema internal. Pemisahan antara skema eksternal dan skema internal berfungsi untuk menjaga bila terjadi perubahan skema internal, misalnya ada penambahan "pointer" pada rekaman tidak memerlukan perubahan pada aplikasi.
- c. Perubahan skema eksternal. Adanya penambahan skema eksternal atau pembuatan skema eksternal baru tidak akan berpengaruh pada aplikasi yang ada selama aplikasi tersebut tidak mengakses data berdasarkan skema yang baru.

## Komponen DBMS

Komponen-komponen DBMS (Howe,1991) terdiri dari:

- Interface, yang didalamnya terdapat bahasa manipulasi data (*data manipulation language*)
- Bahasa definisi data (*data definition language*) untuk skema eksternal, skema konseptual dan skema internal.
- Sistem kontrol basis data (*Database Control System*) yang mengakses basis data karena adanya perintah dari bahasa manipulasi data.

Contoh bahasa menggunakan komponen-komponen tersebut adalah SQL (*Structured Query Language*). SQL merupakan bahasa standar yang digunakan oleh kebanyakan aplikasi-aplikasi DBMS.

## Klasifikasi DBMS

Sistem Basisi Data dapat diklasifikasikan menjadi tiga bagian, yang terdiri dari:

- a. Klasifikasi berdasarkan model data. Klasifikasi ini terdiri dari model data hirarki, model data jaringan, model data relasional.

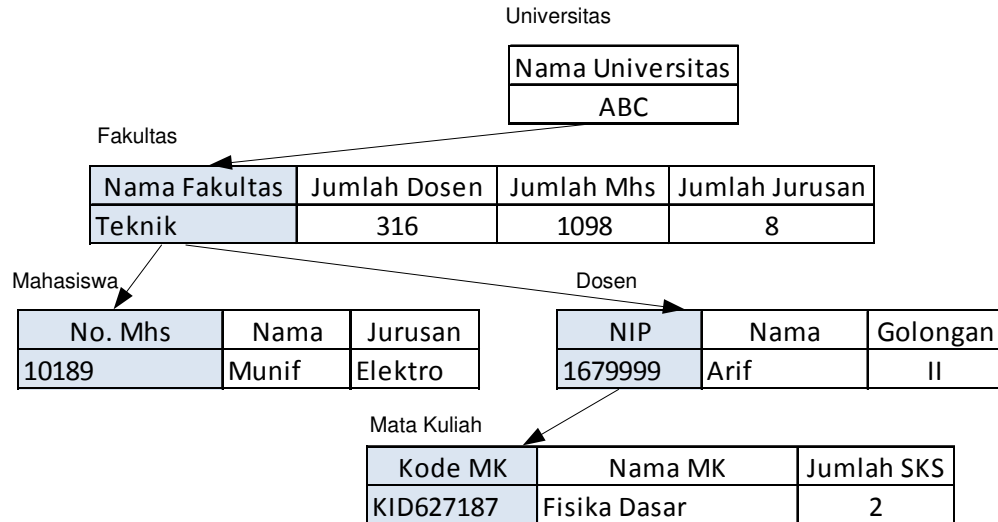
1. Model data hirarki

Dalam model ini, data disusun menurut struktur pohon yang merupakan bentuk lain dari abstraksi data untuk basis data akademi. Pada puncak hirarki disebut dengan akar (*root*). Tiap entitas tingkat atas (*parent*) mempunyai satu atau lebih sub-entitas (*children*) sehingga setiap entitas hanya boleh mempunyai satu induk, tetapi dapat mempunyai banyak anak.

Pada mode data hirarki, hubungan antar entitas dinyatakan dalam satu-banyak (*one to many*) atau satu-satu (*one to one*). Dalam satu Universitas terdapat banyak Fakultas dan setiap Fakultas terdapat banyak Dosen atau banyak Mahasiswa, dan seterusnya. Tanda panah menunjukkan derajat keterhubungan "banyak".

Untuk menampilkan semua mata kuliah pada Fakultas tertentu harus dilakukan dalam dua tahap. Yang pertama adalah menampilkan rekaman semua Dosen yang mengajar di Fakultas tersebut, kemudian baru mata kuliah yang dipegang oleh para Dosen. Dalam hal ini penampilan data terlihat kurang efisien, sebab menggunakan entitas perantara (dosen) yang harus ditampilkan juga. Dikarenakan kunci data yang digunakan untuk menghubungkan antar entitas diberi kode dalam struktur data, maka untuk jumlah entitas perantara yang sedikit masih dapat dikatakan efisien.

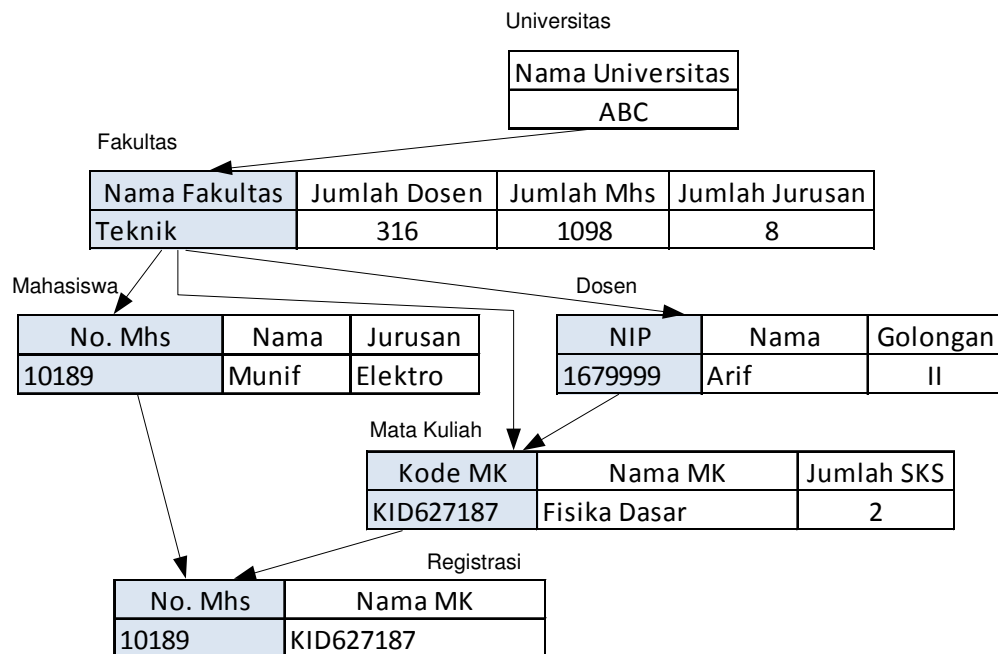
Kelemahan lain pada model data hirarki adalah tidak dapat melakukan pencarian data pada field. Misalnya dalam entitas mata kuliah tidak dapat ditampilkan hanya mata kuliah dengan jumlah SKS tertentu, sebab field "Jumlah SKS" bukan sebagai kunci data. Hal ini masih dapat dilakukan dengan mengubah struktur data dengan memberi hubungan khusus yang digunakan untuk mengubah struktur database. Kelebihan model ini adalah sangat mudah dipahami dan mudah dalam pembaharuan data [Waliyanto2000].



**Gambar 1.4 Organisasi rekaman data pada model hirarki [Waliyanto2000]**

2. Model data Jaringan

Dalam model ini setiap entitas dapat mempunyai banyak induk dan banyak anak. Pada gambar menunjukkan entitas mata kuliah mempunyai dua induk, yaitu langsung berhubungan dengan Fakultas dan Dosen.



**Gambar 1.5 Organisasi rekaman data model raringan [Waliyanto2000]**

Dalam model ini lebih sedikit terdapat data rangkap, namun lebih banyak terdapat hubungan antar entitas, sehingga akan menambah informasi hubungan yang harus disimpan dalam *database*. hal ini akan menambah volume dan kerumitan dalam penyimpanan berkas data.

3. Model data Relasional

Dalam model ini setiap field dapat dijadikan kunci data. Data rekaman disusun dari nilai yang berhubungan (*record*). Baris-baris ini akan membentuk tabel yang umunya tersimpan dalam satu berkas (*file*).

Mata Kuliah				Fakultas			
No MK	Nama MK	SKS	NIP	Fakultas	Jml_Dsn	Jml_Mhs	Jml_Jur
KDI1892	Bahasa Inggris	2	1222	Teknik	150	6320	3
KAI6522	Fisika Lanjut	2	2344	Ekonomi	200	4120	2
...	...	...	...	...	...	...	...

Mahasiswa				Dosen			Registrasi	
No Mhs	Nama	Jurusan	Fakultas	NIP	Nama	Gol	No MK	No Mhs
11782	Arif	Elektro	Teknik	23666	Murdiansyah	III	KDI1892	11782
44366	Rudian	Mesin	Teknik	132455	Johan	II	KAI6522	44366
89878	Nita	Sipil	Teknik	...	...	...	KAI6522	89878
...	...	...	...	...	...	...	...	...

**Gambar 1.6 Organisasi basis data model relasional [Waliyanto2000]**

Dengan menggunakan model ini, pencarian field dari suatu tabel atau banyak tabel dapat dilakukan dengan cepat. Pencarian atribut yang berhubungan pada tabel yang berbeda dapat dilakukan dengan menghubungkan terlebih dahulu tabel-tabel tersebut dengan menggunakan atribut yang sama (*joint operation*).

Keuntungan yang didapat dengan menggunakan model ini adalah sebagai berikut [Waliyanto2000]:

- Model ini lebih luwes karena nilai data dalam tabel tidak ada pembatasan dalam berbagai proses pencarian data.
- Model ini mempunyai latar belakang teori matematika.
- Pengorganisasian model relasional sangat sederhana, sehingga mudah dipahami.
- Basis data yang sama biasanya dapat disajikan dengan lebih sedikit terjadi data rangkap (*redudancy data*).

Sedangkan beberapa kelemahan model ini adalah [Waliyanto2000]:

- Lebih sulit dalam implementasinya terutama untuk data dengan jumlah yang besar dan tingkat kompleksitasnya tinggi.
  - Proses pencarian informasi lebih lambat, karena beberapa tabel tidak dihubungkan secara fisik. Dalam manipulasi data yang menggunakan beberapa tabel akan memerlukan waktu yang lama, karena tabel-tabel harus dihubungkan terlebih dahulu.
- b. Klasifikasi berdasarkan lokasi penyimpanan data, yaitu DBMS terpusat dan DBMS terdistribusi. Dalam DBMS terpusat basis data disimpan dalam satu komputer media penyimpan sehingga pengguna sistem mengakses data dari pusat. DBMS terdistribusi, basis data tersebar pada penyimpanan tiap terminal pengguna (client). Antar pengguna dapat mengakses data secara langsung tanpa perlu melalui pusat penyimpanan. DBMS ini memerlukan sistem kontrol yang rumit.
- c. Klasifikasi berdasarkan tujuan DBMS digunakan yaitu tujuan umum (*general purpose*) dan tujuan khusus (*special purpose*). Untuk tujuan umum dapat digunakan untuk berbagai tujuan dengan memperlakukan data sama menurut penggunaannya contoh aplikasinya adalah DBASE, ORACLE, FOXBASE dan sebagainya. DBMS tujuan khusus dirancang dan digunakan untuk keperluan tertentu, sebagai contoh pengelolaan data karyawan pada perusahaan Asuransi.

## Pengembangan Database

Database diproses oleh DBMS untuk digunakan oleh pengembang maupun pengguna, yang mengakses DBMS secara langsung atau tidak langsung melalui program-program aplikasi. Database terdiri dari empat elemen utama yaitu data pengguna, metadata, indeks dan metadata aplikasi [David2002].

### Data Pengguna

Hampir semua database me-representasikan data pengguna sebagai relasi dengan menganggapnya sebagai tabel data. Kolom dalam tabel berisi field-field atau atribut dan baris tabel berisi record/tuple (rekaman) untuk keterangan entitas dalam lingkungan bisnis. Tidak semua relasi diperlukan, beberapa relasi lebih baik distrukturkan dengan proses normalisasi.

Relasi ini dapat digambarkan dengan bentuk hubungan antara pelajar dengan guru sebagai berikut:

**Tabel 1-1 Relasi Pelajar dengan Guru (R1)**

NamaPelajar	TeleponPelajar	NamaGuru	TeleponGuru
Aminudin	7778889	Pardi	7789665
Usman	7896532	Pardi	7789665
Ari	7474856	Dadang	8965555
Rina	7895654	Marni	4562211
Tuti	7897744	Dadang	8965555
Joni	7845644	Dadang	8965555

Struktur relasi tersebut dapat terjadi beberapa masalah, misalnya jika guru Dadang mengganti nomor telepon maka tiga record yang terdapat guru Dadang diatas harus diganti juga. Untuk itu lebih baik jika struktur relasi diubah menjadi dua relasi seperti di bawah ini:

**Tabel 1-2 Hubungan antara R1 dan R2**

NamaPelajar	TeleponPelajar	NamaGuru
Aminudin	7778889	Pardi
Usman	7896532	Pardi
Ari	7474856	Dadang
Rina	7895654	Marni
Tuti	7897744	Dadang
Joni	7845644	Dadang

NamaGuru	TeleponGuru
Pardi	7789665
Dadang	8965555
Marni	4562211

Dari relasi diatas akan pengubahan data hanya dilakukan pada relasi kedua.

### Metadata

Penjelasan struktur dari suatu tabel disebut dengan *metadata* dan terkadang disebut dengan *system tables*. Bentuk dari metada dapat digambarkan seperti dibawah ini yang terdiri dua tabel. Tabel pertama berisi daftar tabel-tabel di dalam suatu database sedangkan tabel yang kedua berisi daftar kolom-kolom pada suatu tabel.

**Tabel 1-3 Tabel SysTable**

Nama Tabel	Jumlah Kolom	Primary Key
Pelajar	4	NIS
Guru	3	NIP
Mata Pelajaran	4	Kode_MP
Relasi Belajar	3	{NIS,Kode_MP,NIP}

**Tabel 1-4 Tabel SysColumns**

Nama Kolom	Nama Tabel	Tipe Data	Panjang
NIS	Pelajar	String	5
Nama	Pelajar	String	20
Telepon	Pelajar	String	12
Alamat	Pelajar	String	50
NIP	Guru	String	6
Nama	Guru	String	20
Telepon	Guru	String	12
Divisi	Guru	String	20
Kode_MP	Mata Pelajaran	String	5
Nama MP	Mata Pelajaran	String	15
Jumlah Jam	Mata Pelajaran	Integer	4
NIS	Relasi Belajar	String	5
Kode_MP	Relasi Belajar	String	5
NIP	Relasi Belajar	String	6
Tingkat	Relasi Belajar	String	2

### Indeks

Tipe database ini digunakan untuk meningkatkan kinerja dan akses suatu database. Terkadang tipe data ini disebut dengan *overhead data*, terdiri dari prinsip-prinsip indeks serta beberapa penggunaan struktur data *link list*. Di bawah ini contoh penggunaan dua buah indeks dari tabel Mahasiswa:

**Tabel 1-5 Contoh Tabel Mahasiswa**

NO	Nama	Jurusan	Kelas
10	David Carradine	Akuntansi	2AB
20	Jaka Sembung	Manajemen	2CV
30	Kebo Ireng	Manajemen	2CV
40	Lasmini	Teknik Sipil	1SP
50	Joni Keboy	Akuntansi	1AB
60	Franc De Nero	Manajemen	2AB
70	Marco Van Basten	Teknik Sipil	1SP
80	Maradani	Teknik Sipil	1SP
90	Dona Doni	Akuntansi	1AB

**Tabel 1-6 Tabel Indeks berdasarkan Kelas**

Kelas	No
1AB	50,90
2AB	10,60
2CV	20,30
1SP	40,70,80

**Tabel 1-7 Tabel indeks berdasarkan Jurusan**

Jurusan	No
Akuntansi	10,50,90
Manajemen	20,30,60
Teknik Sipil	40,70,80



Indek tidak hanya digunakan untuk pengurutan, tetapi digunakan juga untuk mengakses cepat ke database terutama pencarian data. Apabila suatu tabel contohnya tabel Mahasiswa, mengalami perubahan data (penambahan/pengubahan/penghapusan) maka tabel indeks mengalami perubahan juga.

### Application Metadata

Application metadata digunakan untuk menyimpan struktur dan format dari *user forms, report, queries* dan komponen-komponen aplikasi lainnya.

### Konsep Dasar Tabel

Tabel merupakan blok dasar yang paling umum digunakan dalam sistem basis data, atau disebut juga dengan relasi. Komponen tabel terdiri dari beberapa kolom yang ditandai dengan **jenis atribut**. Perpotongan antara baris dan kolom disebut **nilai atribut**. Tujuan penggunaan tabel adalah untuk menyederhanakan logika pandangan terhadap data. Beberapa ketentuan-ketentuan dalam penyusunan sebuah tabel adalah sebagai berikut [Waliyanto2000]:

- Urutan baris diabaikan, sehingga pertukaran baris tidak berpengaruh pada isi informasi tabel.
- Urutan kolom diabaikan serta identifikasi kolom dibedakan dengan jenis atribut.
- Tiap perpotongan antara baris dan kolom berisi atribut tunggal
- Tiap baris dalam tabel harus dibedakan, sehingga tidak ada dua baris atau lebih dalam tabel mempunyai nilai atribut yang sama secara keseluruhan.

Tabel yang memenuhi ketentuan ini disebut dengan tabel normal, jika belum maka dilakukan proses normalisasi.

Salah satu keuntungan menggunakan basis data adalah konsistensi data selalu terjaga dengan menghindari adanya data rangkap (*redundant data*). Perbedaan antara data rangkap dan data duplikat adalah duplikasi data terjadi bila satu atribut mempunyai dua atau lebih nilai yang sama, sedangkan data rangkap adalah bila satu atribut mempunyai dua atau lebih nilai yang sama, namun bila salah satu nilai dihapus, maka tidak ada informasi yang hilang, sehingga duplikasi data ini tidak perlu ada. Untuk lebih jelasnya lihat dua tabel berikut:

**Tabel 1-8 Contoh duplikasi data pada suatu tabel**

NamaPelajar	TeleponPelajar	NamaGuru
Aminudin	7778889	Pardi
Usman	7896532	Pardi
Ari	7474856	Dadang
Rina	7895654	Marni
Tuti	7897744	Dadang
Joni	7845644	Dadang

**Tabel 1-9 Contoh adanya kerangkapan data pada suatu tabel**

NamaPelajar	TeleponPelajar	NamaGuru	TeleponGuru
Aminudin	7778889	Pardi	7789665
Usman	7896532	Pardi	7789665
Ari	7474856	Dadang	8965555
Rina	7895654	Marni	4562211
Tuti	7897744	Dadang	8965555
Joni	7845644	Dadang	8965555

Pada tabel 1.8 terjadi duplikasi data pada atribut NamaGuru, andaikan baris pertama pada atribut NamaGuru dihilangkan maka informasi untuk atribut NamaPelajar baris pertama akan hilang, sedangkan pada tabel 1.9 dapat terlihat bahwa kalau atribut TeleponGuru dari baris pertama dihilangkan maka informasi ini masih dapat diketahui melalui atribut NamaGuru pada baris kedua, mengapa?

Salah satu syarat tabel normal adalah setiap atribut harus mempunyai nilai tunggal untuk tiap barisnya. Di bawah ini contoh dari suatu tabel yang mempunyai atribut bernilai ganda.

**Tabel 1-10 Tabel Dosen dengan nilai ganda**

NIP	Nama	Gelar
102	Jackie Ceng	Ir
106	Dakocan	Ir, MSc
503	Ali Oncom	Drs
401	Otoy	Ir, MSc, PhD
203	Gareng	Prof, Drs

Dalam tabel di atas terdapat nilai atribut ganda pada kolom Gelar. Hal ini berakibat pengurutan data hanya dapat dilakukan berdasarkan kolom NIP dan Nama. Untuk menghilangkan nilai ganda tersebut, hal yang paling mudah dilakukan adalah membuat pengisian nilai atribut vertikal namun dapat berakibat kerangkapan data, seperti di bawah ini.

**Tabel 1-11 Pengisian atribut secara vertikal**

NIP	Nama	Gelar
102	Jackie Ceng	Ir
106	Dakocan	Ir
106	Dakocan	MSc
503	Ali Oncom	Drs
401	Otoy	Ir
401	Otoy	MSc
401	Otoy	PhD
203	Gareng	Prof
203	Gareng	Drs

Solusi yang tepat untuk menghilangkan kerangkapan data tersebut adalah dengan membagi tabel menjadi dua bagian yang saling terhubung dengan elemen penghubung salah satu atributnya. Perhatikan tabel-tabel di bawah ini:

**Tabel 1-12 Menghilangkan nilai rangkap dengan pemecahan tabel**

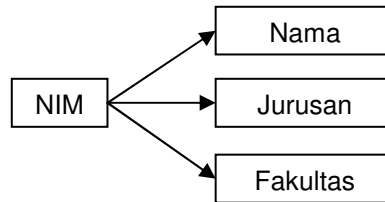
		Gelar	
NIP	Nama	NIP	Gelar
102	Jackie Ceng	102	Ir
106	Dakocan	106	Ir
503	Ali Oncom	106	MSc
401	Otoy	503	Drs
203	Gareng	401	Ir
		401	MSc
		401	PhD
		203	Prof
		203	Drs

Dengan cara ini dapat mempermudah dalam proses normalisasi berikutnya. Dalam penyusunan aturan data perlu dipahami tentang **determinan** dan **identitas**. Jika sebuah tabel memiliki atribut A, B, dan C, sedangkan A menjadi penentu B atau sebaliknya B ditentukan oleh A maka A determinan (*functional determines*) B (B *functional dependent* A) . Nilai atribut B dapat saja duplikasi, kosong atau dapat diubah. Jika  $a_1$  dan  $b_1$  merupakan nilai A maka akan berpasangan dengan nilai B yang sama ataupun berbeda. Jadi A determinan B jika tiap A mempunyai satu pasangan nilai B. Perhatikan contoh tabel di bawah ini:

**Tabel 1-13 Tabel Mahasiswa**

NIM	Nama	Jurusan	Fakultas
21297956	Asmawi	Teknik Sipil	Teknologi Industri
21297556	Tina	Manajemen	Ekonomi
20399458	Marino	Akuntansi	Ekonomi
21198113	Budi	Teknik Sipil	Teknologi Industri

Apabila setiap nilai atribut NIM menentukan nama mahasiswa maka dikatakan atribut NIM determinan atribut Nama. Begitu juga dengan atribut Jurusan dan Fakultas yang ditentukan oleh NIM. Bentuk diagram determinan adalah sebagai berikut:



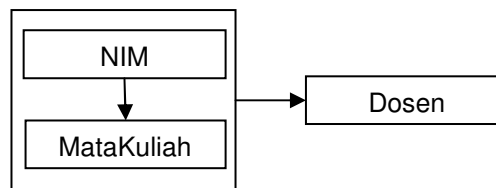
**Gambar 1.7 Diagram determinan dari tabel Mahasiswa**

Dalam kasus lain, ada kemungkinan dua atribut atau lebih secara bersama menentukan atribut lain atau determinan komposit (*composite determinant/fully functionally dependent*). Sebagai contoh pada tabel di bawah, atribut NIM dan atribut MataKuliah menentukan atribut Dosen sebagai pengajar.

**Tabel 1-14 Tabel pengajaran mata kuliah**

NIM	MataKuliah	Dosen
21297956	Matematika Dasar	Pardi
21297956	Fisika Dasar	Munir
20399458	Matematika Dasar	Joko Susilo
21198113	Fisika Dasar	Munir
20399458	Akuntansi Dasar	Marni Siregar
21297556	Matematika Dasar	Joko Susilo

Gambar diagram determinannya adalah sebagai berikut:

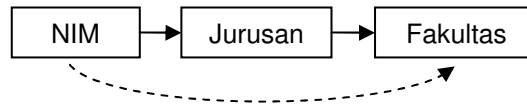


**Gambar 1.8 Diagram determinan tabel pengajaran mata kuliah**

Sedangkan bila atribut A determinan atribut B dan atribut B merupakan determinan atribut C maka atribut A adalah determinan transitif atribut C (*C transitive dependency A*). perhatikan contoh tabel dan diagram determinan di bawah ini:

**Tabel 1-15 Daftar penerimaan mahasiswa baru**

NIM	Jurusan	Fakultas
21297956	SI	ILKOM
21297556	SK	ILKOM
20399458	TI	TI
21198113	SK	ILKOM



**Gambar 1.9 Diagram A determinan transitif C**

Dari pembahasan di atas tiap baris dapat diidentifikasi dengan semua nilai atribut, tetapi akan sangat menyulitkan. Oleh sebab itu perlu pemilihan salah satu nilai atribut yang digunakan sebagai identitas (*identifier*) atau elemen kunci (*key element*) dari baris. Nilai atribut dapat dijadikan identitas jika dalam tabel tidak terjadi duplikasi data dan data dengan nilai kosong (NULL).

### Normalisasi

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data diantara relasi dalam perancangan logikal basis data. Format normalisasi terdiri dari lima bentuk, yaitu:

**Form Normal Pertama (1NF).** Suatu tabel dikatakan sudah 1NF jika telah memenuhi ketentuan sebagai berikut:

- Tidak ada atribut mempunyai nilai berulang atau nilai array
- Tidak mempunyai baris yang rangkap

Bentuk unnormal mengijinkan nilai-nilai pada suatu atribut dapat berulang. Perhatikan contoh tabel-tabel berikut ini: [Sitansu1991]

**Tabel 1-16 Tabel UNIV (University)**

DNO	DNAME	DHEAD
EN	English	Lee Kunkel
CS	Computer Science	Albert Roby
MA	Mathematics	Deb Kumar Boy
HS	History	Cathy Doucette
EE	Electrical Engineering	Raj Chandra Mittra

**Tabel 1-17 Tabel INSTR (Instructor)**

INAME	IDEG	SPCODE	RANK	SSNO	DNO
Lee Kunkel	BA, MA, PhD	4	Professor	2323121	EN
Albert Roby	BS, MS, PhD	2	Professor	1212154	CS
Deb Kumar Boy	BS, MS, PhD	5	AssocProf	4545454	MA
Cathy Doucette	MA, PhD	6	AssocProf	4545654	HS
Raj Chandra Mittra	BA, MSc, PhD	10	Professor	2231321	EE
Tom Clark	BA, MA	5	AsstProf	1546465	MA
Marcia Brown	BA, BS, MS	2	Instructor	4464654	CS
Susan Woodsmith	MA, MS, PhD	3	AsstProf	2131321	PH
Brady Jackson	MA, DLitt	15	Professor	2456465	RL
Jack Adams	BA, PhD	1	AssocProf	4545462	CS

**Tabel 1-18 Tabel STUDNT (Student)**

SNAME	SSNO	MAJOR	DEGREE	ADVSR	DNO	COLREG
Roger Brown Smith	121545	Biology	BS	Jack Adams	BI	Arts & Sci.
Cindy Logan	232332	Computer Science	BS	Deb Kumar Boy	MA	Arts & Sci.
Benjamin Johnson	554545	NDEG	NONE		BA	Business
Steve Levin	454545		BA	Lee Kunkel	EN	Arts & Sci.
Tom Jones	899778	Mathematics	MS	Raj Chandra Mittra	EE	Engineering
Beverly Black	365654	English	PhD	Lee Kunkel	EN	Arts & Sci.

**Tabel 1-19 Tabel CRSE (Course)**

CNO	CNAME	INAME	DNO	SECNO
CS225	Assembler Language	Marcia Brown	CS	02
CS547	Discrete Mathematics	Deb Kumar Roy	CS	01
MA423	Differential Geometry	Tom Clark	MA	04
EN104	English Composition	Staff	EN	04
RL712	Comparative Religion	Brady Jackson	RL	01
CS761	Expert System	Albert Roby	CS	03
EC102	Macroeconomics	Staff	EC	06
EN604	Romanticism	Lee Kunkel	EN	01
HS252	Middle East	Cathy Doucette	HS	02
EE202	Microcomputing	Staff	EE	04
MA611	Algebraic Topology	Tom Clark	MA	01
CS579	Database Systems	Marcia Brown	CS	02
BI104	Biology Concepts	Staff	BI	07

**Tabel 1-20 Tabel CRSLST (Course List)**

CNO	SECNO	SID	GRADE	OFRNG
CS579	02	121212	A	Spring 87
CS579	02	121231	B-	Spring 87
CS579	02	454549	B+	Spring 87
CS579	02	484545	I	Spring 87
MA611	01	112121	C	Fall 86
MA611	01	212121	A	Fall 86
MA611	01	545454	C+	Fall 86
MA611	01	121215	W	Fall 86

**Tabel 1-21 Tabel SPECL (Special)**

SPCODE	SPNAME
1	Information Systems
2	Database Systems
3	Kant Doctrine
4	Romantic Literature
5	Differential Geometry
6	Mideast History
7	Topology
8	Automated Reasoning
9	Expert System
10	Microelectronics
11	English Drama
12	Shakespeare
13	Indian History
14	Decision Support Systems
15	Comparatibe Religion

Tabel UNIV, STUDNT, CRSE, CRELST dan SPCODE semuanya berada dalam 1NF, namun untuk tabel INSTR masih dalam keadaan unnormalisasi, karena atribut IDEG menerima nilai array seperti "BA, MA, PhD" atau "MA, MS, PhD".

**Form Normal Kedua (2NF).** Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

- Key terdiri dari atribut tunggal
- Setiap atribut nonkey ketergantungan fungsional pada semua key atau tidak terjadinya ketergantungan pada key *composite*.

Misalnya tabel UNIV berada dalam normal kedua dengan mengasumsikan DNO sebagai key, kecuali CRSE. Jika ditentukan CNO dan SECNO sebagai key composite, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

**Form Normal Ketiga (3NF).** Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada *ketergantungan transitif* diantara atribut. Misalnya tabel STUDNT mempunyai atribut SSNO sebagai key (2NF). *Ketergantungan transitif* terjadi diantara DNO dan COLREG. Saat DNO determinan COLREG tanpa melibatkan key SSNO. Contohnya, DNO='CS' termasuk COLREG='Arts/Sc.' tidak tergantung oleh atribut SSNO, sehingga STUDNT belum termasuk 3NF. Yang menjadi catatan, *ketergantungan transitif* tidak akan terjadi jika ada *ketergantungan fungsional* diantara atribut-atribut nonkey yang melibatkan key. Misalnya atribut nonkey SNAME determinan atribut nonkey lainnya yaitu MAJOR, DEGREE, ADVSR dan DNO. Tetapi hal ini merupakan *ketergantungan fungsional* bukan *ketergantungan transitif* selagi semua melibatkan key SSNO.

**Form Normal Boyce-Codd (BCNF).** BCNF menentukan setiap determinan adalah kunci kandidat (*candidate key*). Misalnya UNIV mempunyai dua determinan yaitu DNO dan DNAME yang merupakan *kunci kandidat* sehingga termasuk ke dalam BCNF. Di lain pihak CRSLST dalam 3NF tetapi tidak dalam BCNF. Atribut komposisinya (CNO, SECNO, SID, OFRNG) sebagai kunci-kunci kandidat dan tidak ada ketergantungan transitif, sehingga CRSLST termasuk ke dalam 3NF. Namun atribut CNO adalah determinan saat SECNO tergantung penuh secara fungsional terhadap CNO, walaupun CNO bukan kunci kandidat, sehingga CRSLST belum termasuk BCNF.

**Form Normal Keempat (4NF).** Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivalued dependency*).

**Form Normal Kelima (5NF).** Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*join dependency*).

Bentuk lain proses normalisasi dapat anda lihat dalam tabel-tabel di bawah ini:

**Tabel 1-22 Tabel Mahasiswa dalam 1NF**

NIM	NamaAwal	NamaAkhir	Fakultas
122233	Asep	Darma	Ilmu Komputer
233323	Angling	Darma	Ilmu Komputer
244455	Bergola	Ijo	Hukum
334343	Jaka	Sembung	Kebidanan
322323	Jaka	Tarub	Hukum

Pada tabel di atas berada pada 1NF kerana tidak ada baris yang duplikat, setiap kolom hanya mempunyai nilai tunggal (tidak ada group berulang atau array) dan semua masukan dalam kolom mempunyai jenis yang sama.

Key NIM secara fungsional menentukan atribut lain seperti NamaAwal, NamaAkhir, dan Fakultas (dengan asumsi, setiap mahasiswa hanya boleh menempatkan satu fakultas). NIM→NamaAwal, NIM→NamaAkhir, NIM→Fakultas.

Atribut key menentukan secara unik nilai dari atribut lain dalam tabel, semua atribut non-key dalam tabel secara fungsional tergantung terhadap key. Tetapi ada kemungkinan atribut non-key dalam tabel dapat menentukan atribut lain pada tabel tersebut. Perhatikan tabel berikut:

**Tabel 1-23 Atribut Non-Key Determinan**

NamaAwal	NamaAkhir	Fakultas	Jenjang
Asep	Darma	Ilmu Komputer	S1
Angling	Darma	Ilmu Komputer	S1
Bergola	Ijo	Hukum	S1
Jaka	Sembung	Kebidanan	D3
Jaka	Tarub	Hukum	S1

Atribut jenjang dapat dikatakan tergantung secara fungsional pada atribut Fakultas dengan konsekuensi bahwa Fakultas Ilmu Komputer dan Hukum hanya untuk mahasiswa S1 dan Fakultas Kebidanan hanya untuk mahasiswa D3.

Pada tabel 1.23 juga memiliki *composite key* (kunci komposisi) yang terdiri dari atribut NamaAwal dan NamaAkhir dan atribut Jenjang tergantung secara fungsional pada *composite key* tersebut. Sebenarnya bisa saja atribut Jenjang tergantung pada atribut NamaAkhir (jika nilai atribut NamaAkhir tidak ada yang sama, namun NamaAkhir terdapat nilai yang sama yaitu "Darma"), atau tergantung dengan atribut NamaAwal, tetapi atribut NamaAwal mempunyai nilai duplikat yaitu "Jaka". Maka atribut Jenjang tidak tergantung fungsional terhadap kedua atribut key tersebut. Sehingga tabel tersebut belum termasuk ke dalam 2NF (semua atribut non-key tergantung pada semua key).

Tabel 1.23 dapat dinormalisasikan ke dalam bentuk 2NF jika di tambahkan atribut NIM, sehingga tabel hanya mempunyai atribut tunggal (non-composite key) seperti di bawah ini.

**Tabel 1-24 Tabel Normalisasi 2NF**

NIM	NamaAwal	NamaAkhir	Fakultas	Jenjang
122233	Asep	Darma	Ilmu Komputer	S1
233323	Angling	Darma	Ilmu Komputer	S1
244455	Bergola	Ijo	Hukum	S1
334343	Jaka	Sembung	Kebidanan	D3
322323	Jaka	Tarub	Hukum	S1

### **Anomali (Pelanggaran) dan Normalisasi**

Untuk mengilustrasikan adanya anomali, anggap terjadi pada atribut Jenjang dari Fakultas, misalnya Kebidanan. Jika Jaka Sembung kembali berkelana (tidak jadi kuliah) maka baris yang berhubungan dengan Jaka Sembung akan dihapus dan terjadi kehilangan informasi bahwa Kebidanan mempunyai jenjang D3. Kita dapat saja mengetahui bahwa Jenjang Kebidanan adalah D3, tetapi dalam database tersebut tidak ada informasi yang menyatakan Jenjang Kebidanan.

Contoh lain misalnya penambahan mahasiswa baru yang bernama Suparman yang ingin kuliah di Fakultas Teknik. Fakta dari informasi tabel kita tidak dapat mengetahui apa jenjang dari Fakultas Teknik dan juga apakah Fakultas Teknik terdapat pada Universitas yang bersangkutan. Dari hal ini kita tidak akan mengetahui apakah dapat nilai Teknik tersebut dapat dimasukkan ke dalam atribut Fakultas atau tidak. Apabila dapat dimasukkan ke atribut Fakultas, bagaimana dengan atribut Jenjang-nya?

DBMS hanya dapat bekerja dengan informasi yang terdapat pada tabel-tabel dan aturan-aturan yang bekerja pada tabel-tabel tersebut dengan tepat dan mungkin.

Bagaiman hubungan antara anomali dan normalisasi? Jawaban yang singkat adalah dengan menyusun tabel-tabel dalam database cukup dinormalkan (dalam praktek umumnya sampai normal keempat), dan menjamin bahwa anomali tidak terjadi pada database.

Porses normalisasi kelihatan sangat menyulitkan, ketika melihat dari definisi tiap-tiap tingkatan normalisasi. Namun dalam prakteknya kita dapat mencapainya dengan menjamin bahwa tabel-tabel terdiri dari tabel "single-theme".

Walaupun dalam 2NF dapat terjadi penambahan maupun penghapusan data yang mengakibatkan anomali, kita dapat membentuk tabel tersebut menjadi beberapa tabel "single-theme". Gagasan ini dapat diilustrasikan pada tabel di bawah ini.

**Tabel 1-25 Tabel-tabel "Single-Theme"**

NIM	NamaAwal	NamaAkhir	Fakultas	Jenjang	NIM	Fakultas
122233	Asep	Darma	Ilmu Komputer	S1	122233	Ilmu Komputer
233323	Angling	Darma	Hukum	S1	233323	Ilmu Komputer
244455	Bergola	Ijo	Kebidanan	D3	244455	Hukum
334343	Jaka	Sembung			334343	Kebidanan
322323	Jaka	Tarub			322323	Hukum

**Transitive Dependencies (Ketergantungan transitif) (3NF)**

Seperti yang telah kita ketahui, ketergantungan transitif terjadi bilamana suatu  $A \rightarrow B$  dan  $B \rightarrow C$ , maka  $A \rightarrow C$ . Ilustrasi untuk kejadian ini dapat dilihat pada tabel berikut.

**Tabel 1-26 Tabel Daftar Buku**

Author Last Name	Author First Name	Book Title	Subject	Collection or Library	Building
Berdahl	Robert	The Politics of the Prussian Nobility	History	PCL General Stacks	Perry-Casta Library
Yudof	Mark	Child Abuse and Neglect	Legal Procedures	Law Library	Townes Hall
Harmon	Glynn	Human Memory and Knowledge	Cognitive Psychology	PCL General Stacks	Perry-Casta Library
Graves	Robert	The Golden Fleece	Greek Literature	Classics Library	Waggener Hall
Miksa	Francis	Charles Ammi Cutter	Library Biography	Library and Information Science Collection	Perry-Casta Library
Hunter	David	Music Publishing and Collecting	Music Literature	Fine Arts Library	Fine Arts Building
Graves	Robert	English and Scottish Ballads	Folksong	PCL General Stacks	Perry-Casta Library

Dari tabel di atas kita dapat berpendapat bahwa buku yang ber-subyek History, cognitive psychology, dan folksong diberikan ke PCL General Stacks collection; sedangkan Legal procedure diberikan ke Law Library; Greek Literature diberikan ke Classic Library; sedangkan Library Biography diberikan ke Library and Information Science Collection; dan Music Literature diberikan ke Fine Arts Library.

Kemudian kita dapat menduga bahwa PCL General Stacks Collection dan LISC keduanya ditempatkan di gedung Perry-Casta Library (PCL); Classic Library ditempatkan di gedung Waggener Hall; Law Library and Fine Arts Library di tempatkan di gedung Townes Hall dan Fine Arts Building.

Sehingga kita dapat melihat ketergantungan transitif pada tabel di atas, dimana buku-buku History, Cognitive Psychology atau Library Biography secara fisik ditempatkan di gedung PCL; buku Lega Procedures di tempatkan di Townes Hall dan begitu seterusnya.



Apa yang salah ketika terjadinya ketergantungan transitif pada tabel tersebut? Hal pertama terjadinya duplikasi informasi, dimana tiga baris berbeda mengacu pada PCL General Stacks yang berada di gedung PCL. Kesalahan yang kedua adalah kemungkinan adanya penghapusan anomali, yaitu jika kita hapus baris dengan nama pengarang Yudof maka kita akan kehilangan informasi Law Library di dalam Townes Hall. Ketiga dapat pula terjadi penambahan anomali jika kita akan menambah buku Chemistry, tetapi kenyataannya tidak ada data yang menyatakan Chemistry Library berada di gedung Robert A. Welch Hall. Dan masalah keempat terjadinya kesalahan meng-update jika pegawai memasukan buku ke LISC tetapi memasukkan Townes Hall ke atribut Building.

Solusi untuk masalah di atas adalah membentuk tabel tersebut menjadi tabel-tabel "single-theme".

**Tabel 1-27 Tabel-tabel "single-theme " untuk tabel transitive dependencies**

Author Last Name	Author First Name	Book Title	Book Title	Subject
Berdahl	Robert	The Politics of the Prussian Nobility	The Politics of the Prussian Nobility	History
Yudof	Mark	Child Abuse and Neglect	Child Abuse and Neglect	Legal Procedures
Harmon	Glynn	Human Memory and Knowledge	Human Memory and Knowledge	Cognitive Psychology
Graves	Robert	The Golden Fleece	The Golden Fleece	Greek Literature
Miksa	Francis	Charles Ammi Cutter	Charles Ammi Cutter	Library Biography
Hunter	David	Music Publishing and Collecting	Music Publishing and Collecting	Music Literature
Graves	Robert	English and Scottish Ballads	English and Scottish Ballads	Folksong

Subject	Collection or Library	Collection or Library	Building
History	PCL General Stacks	PCL General Stacks	Perry-Casta Library
Legal Procedures	Law Library	Law Library	Townes Hall
Cognitive Psychology	PCL General Stacks	PCL General Stacks	Perry-Casta Library
Greek Literature	Classics Library	Classics Library	Waggenner Hall
Library Biography	Library and Information Science Collection	Library and Information Science Collection	Perry-Casta Library
Music Literature	Fine Arts Library	Fine Arts Library	Fine Arts Building
Folksong	PCL General Stacks	PCL General Stacks	Perry-Casta Library

Kita dapat lihat bahwa semua tabel tidak mempunyai ketergantungan transitif (3NF, Domain Key Normal Form (DKNF)).

Untuk tabel pengarang terdapat nama depan yang sama yaitu Robert, dalam hal ini kita menyarankan untuk membuat atribut identifikasi dalam menyatakan nama pengarang, seperti di bawah ini.

**Tabel 1-28 Tabel pengarang setelah penambahan atribut ID**

Author Last Name	Author First Name	ID Author
Berdahl	Robert	001
Yudof	Mark	002
Harmon	Glynn	003
Graves	Robert	004
Miksa	Francis	005
Hunter	David	006
Graves	Robert	007

ID Author	Book Title
001	The Politics of the Prussian Nobility
002	Child Abuse and Neglect
003	Human Memory and Knowledge
004	The Golden Fleece
005	Charles Ammi Cutter
006	Music Publishing and Collecting
007	English and Scottish Ballads

Pembentukan tabel-tabel di atas akan lebih menghemat kapasitas media penyimpanan dan meminimalkan kesalahan dalam pemasukkan data yang berupa key.

BCNF mengijinkan terjadinya anomali ketika tabel gagal memiliki properti yaitu setiap determinan adalah kunci kandidat (*candidate key*). Contoh pada tabel di bawah ini gagal memiliki properti ini. Dalam tabel ini SSN ditafsirkan sebagai pelajar dengan Major serta Adviser (pembimbing). Dengan catatan tiap pelajar pelajar 123-45-6789 dan 987-65-4321 mempunyai dua jurusan dengan pembimbing yang berbeda tiap jurusannya.

**Tabel 1-29 Tabel Mahasiswa dengan BCNF**

SSN	Major	Adviser
123-45-6789	Library and Information Science	Dewey
123-45-6789	Public Affairs	Roosevelt
222-33-4444	Library and Information Science	Putnam
555-12-1212	Library and Information Science	Dewey
987-65-4321	Pre-Medicine	Semmelweis
987-65-4321	Biochemistry	Pasteur
123-54-3210	Pre-Law	Hammurabi

Dalam tabel di atas salah satu determinan adalah atribut berpasangan yaitu SSN dan Major. Tiap pasangan nilai atribut SSN dan Major menentukan nilai unik untuk atribut Adviser. Determinan lain adalah pasangan SSN dan Adviser yang menentukan nilai unik atribut Major. Determinan lain lagi adalah Adviser dimana setiap nilai adviser yang berbeda menentukan secara unik nilai atribut Major. (catatan: setiap pelajar mempunyai pembimbing (adviser) tunggal untuk setiap jurusannya (major) dan tiap pembimbing (adviser) membimbing hanya satu jurusan)

Sekarang kita akan menguji ketiga determinan tersebut apakah sebagai kunci kandidat. Pasangan SSN dan Major adalah kunci kandidat untuk setiap pasangan unik mengidentifikasi sebuah baris dalam tabel. Pasangan SSN dan Adviser juga kunci kandidat. Tetapi determinan Adviser bukan sebagai kunci kandidat karena nilai Dewey terdapat dalam dua baris pada kolom Adviser. Jadi tabel di atas tidak dapat dikatakan dalam kondisi setiap determinan adalah kunci kandidat.

Sangatlah mudah untuk memeriksa anomali dalam tabel ini. Misalnya jika pelajar 987-65-4321 meninggalkan universitas tersebut, tabel akan kehilangan informasi bahwa Semmelweis adalah seorang pembimbing untuk jurusan Pre-Medicine. Contoh lain adalah tidak adanya informasi tentang pembimbing-pembimbing untuk pelajar dengan jurusan History (Sejarah).

Solusi untuk masalah di atas adalah membuat tabel tersebut dipecah menjadi beberapa tabel-tabel single-theme seperti di bawah ini:

SSN	Adviser	Major	Adviser
123-45-6789	Dewey	Library and Information Science	Dewey
123-45-6789	Roosevelt	Public Affairs	Roosevelt
222-33-4444	Putnam	Library and Information Science	Putnam
555-12-1212	Dewey	Pre-Medicine	Semmelweis
987-65-4321	Semmelweis	Biochemistry	Pasteur
987-65-4321	Pasteur	Pre-Law	Hammurabi
123-54-3210	Hammurabi	History	Herodotus

Maka kedua tabel di atas dalam kondisi BCNF yaitu setiap determinan merupakan kunci kandidat.

Bentuk formal keempat memperhatikan terjadinya anomali ketika tabel gagal memiliki properti bahwa tidak ada ketergantungan nilai banyak (multivalued). Contoh tabel di bawah merupakan bentuk anomali untuk ketergantungan dengan nilai banyak.

**Tabel 1-30 Mahasiswa dengan daftar hobi**

LastName	Major	Hobby
Jones	Library and Information Science	Surfing the Internet
Jones	Library and Information Science	Chess
Jones	Public Affairs	Surfing the Internet
Jones	Public Affairs	Chess
Lee	Library and Information Science	Photography
Lee	Library and Information Science	Stamp collecting
Ruiz	Pre-Medicine	Surfing the Internet
Ruiz	Pre-Medicine	Photography
Ruiz	Biochemistry	Surfing the Internet
Ruiz	Biochemistry	Photography
Smith	Pre-Law	Playing poker

Anggap pelajar Jack Jones hobinya menjelajah internet dan bermain catur; Lynn Lee hobinya photographer dan koleksi perangko; Mary Ruiz, hobinya menjelajah internet dan photographer; dan Lynn Smith, hobinya bermain poker.

Situasi ini dapat mengakibatkan redudansi data dan memberikan efek ketergantungan nilai banyak terjadi ketika (a) tabel sekurangnya mempunyai tiga atribut (b) dua atribut mempunyai nilai banyak dan (c) nilai dari atribut multivalued tergantung hanya satu dari sisa atribut. Atribut LastName menentukan nilai banyak dari atribut Major dan Hobby, tetapi keduanya tidak saling tergantung terhadap yang lainnya (independent).

Notasi untuk ketergantungan multivalued adalah panah ganda. Untuk contoh ini kita dapat menulisnya dengan bentuk LastName ⇒ Major dan LastName ⇒ Hobby (LastName multideterminan Major dan LastName Multideterminan Hobby).

Bentuk tabel-tabel yang dijadikan single-theme adalah sebagai berikut:

LastName	Major	LastName	Hobby
Jones	Library and Information Science	Jones	Surfing the Internet
Jones	Public Affairs	Jones	Chess
Lee	Library and Information Science	Lee	Photography
Ruiz	Pre-Medicine	Lee	Stamp collecting
Ruiz	Biochemistry	Ruiz	Surfing the Internet
Smith	Pre-Law	Ruiz	Photography
		Smith	Playing poker

Normal Form Kelima sulit untuk diilustrasikan dengan contoh sederhana. Karenanya di sini tidak meng-ilustrasikan properti dari 5NF mempunyai tiap *join dependency* dalam tabel menjadi penyebab membentuk kunci kandidat tabel. Alasan lain yang pertama, dalam prakteknya 4NF sudah mencukupi dan yang kedua adalah Domain-Key Normal Form (DKNF) sudah termasuk 5NF.

DKNF digunakan sebagai solusi untuk menghindari anomali: kumpulan tabel-tabel (relasi) dalam DKNF sebagai konsekuensi dari teori Ronald Fagin (1981) untuk bebas dari anomali. Definisi dari DKNF adalah relasi dalam DKNF jika setiap *constrain* relasi logikal mendefinisikan kunci dan domain. Menurut Fagin istilah kunci adalah primary key dan candidate key, domain merupakan kumpulan definisi dari isi atribut (kolom) dan beberapa batasan jenis data yang akan disiman dalam kolom, sebagai contoh batasan hanya data numerik atau hanya data logikal. Constraint berarti aturan-aturan terhadap atribut dengan jelas sehingga dapat diputuskan aturan yang mana dijalankan atau dilanggar untuk kumpulan data yang mana diterima. Sebagai ilustrasi ini simak tabel di bawah ini:

**Konversi tabel dengan partial dependency ke tabel DKNF**

**Tabel 1-31 Tabel dengan partial dependency**

FirstName	LastName	Major	Level
Jack	Jones	LIS	Graduate
Lynn	Lee	LIS	Graduate
Mary	Ruiz	Pre-Medicine	Undergraduate
Lynn	Smith	Pre-Law	Undergraduate
Jane	Jones	LIS	Graduate

Dalam tabel di atas terdapat komposisi kunci yang terdiri dari pasangan atribut LastName-FirstName, dan semua atribut tergantung dengan kunci ini. Tetapi ada hal lain yang penting: atribut Level tergantung pada atribut LastName, Level hanya tergantung pada sebagian kunci. Sehingga tabel ini belum dikatakan DKNF.

Kita akan membutuhkan tabel yang menyediakan link antara pasangan atribut FirstName dan LastName dan atribut Major. Dalam tabel tersebut Major menjadi konsekuensi logikal dari key dan domain. Sehingga dibutuhkan dua tabel, satu berisi Major dan Level dan yang lain FirstName, LastName dan Major.

Major	Level	FirstName	LastName	Major
LIS	Graduate	Jack	Jones	LIS
Pre-Medicine	Undergraduate	Lynn	Lee	LIS
Pre-Law	Undergraduate	Mary	Ruiz	Pre-Medicine
		Lynn	Smith	Pre-Law
		Jane	Jones	LIS

Contoh tabel-tabel di bagian sebelumnya sudah termasuk DKNF. Jadi dengan membentuk tabel menjadi single-theme hampir sama halnya dengan membentuk tabel DKNF.

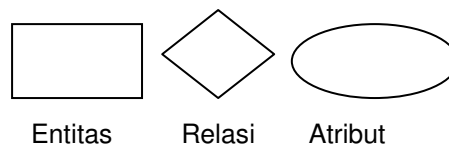
## PEMODELAN BASIS DATA

### **Model Hubungan Antar Entitas (*Entity Relationship-Model*)**

Model entity-relationship pertama kali diperkenalkan oleh Peter Chen pada tahun 1976. Dalam pemodelan ini dilakukan dengan tahapan sebagai berikut:

- Memilih entitas-entitas yang akan disusun dalam basis data dan menentukan hubungan antar entitas yang telah dipilih.
- Melengkapi atribut-atribut yang sesuai pada entitas dan hubungan sehingga diperoleh bentuk tabel normal penuh (ternormalisasi).

Elemen-elemen dalam model ER dapat digambarkan pada gambar diagram di bawah ini:



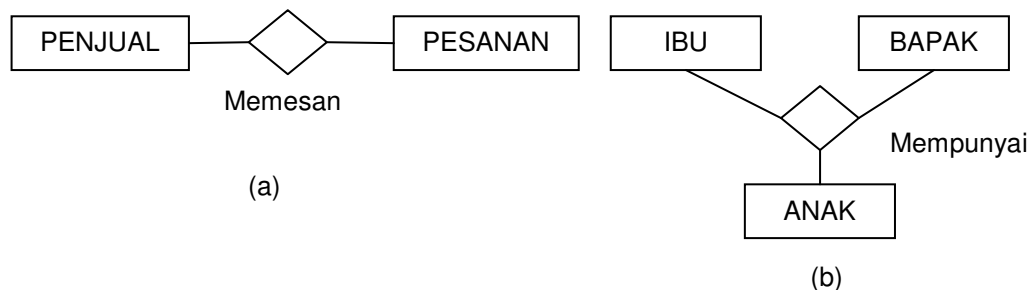
**Gambar 2.1 Elemen-elemen ER-Model**

Entitas merupakan sesuatu yang dapat diidentifikasi dalam lingkungan kerja pengguna. Entitas yang diberikan tipe dikelompokkan ke kelas entitas. Perbedaan antara kelas entitas dan instansi entitas adalah sebagai berikut:

- Kelas entitas adalah kumpulan entitas dan dijelaskan oleh struktur atau format entitas di dalam kelas.
- Instansi kelas merupakan bentuk penyajian dari fakta entitas.

Umumnya terdapat banyak instansi entitas di dalam setiap entitas kelas. Setiap entitas kelas memiliki atribut yang menjelaskan karakteristik dari entitas tersebut, sedangkan setiap instansi entitas mempunyai identifikasi yang dapat bernilai unik (mempunyai nilai yang berbeda untuk setiap identifikasinya) atau non-unik (dapat bernilai sama untuk setiap identifikasinya).

Antara entitas diasosiasikan dalam suatu hubungan (*relationship*). Suatu relasi dapat memiliki beberapa atribut. Jumlah kelas entitas dalam suatu relasi disebut **derajat relasi**. Gambar di bawah ini merupakan contoh dari relasi berderajat dua dan relasi berderajat tiga.

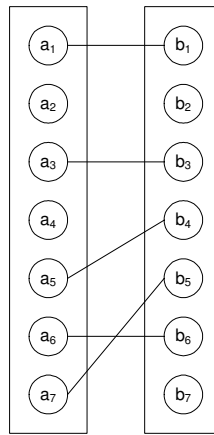


**Gambar 2.2 (a) Relasi berderajat dua (b) Relasi berderajat tiga**

### **Tipe Binary Relationship**

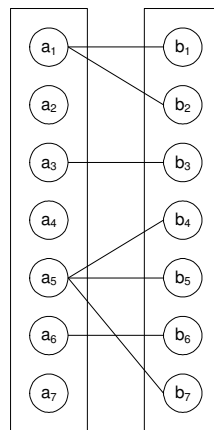
Relasi memiliki tiga tipe biner yaitu:

- One-to-one (1:1). Hubungan terjadi bila setiap instansi entitas hanya memiliki satu hubungan dengan instansi entitas lain.



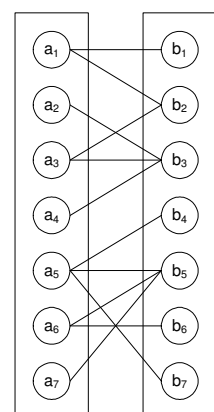
**Gambar 2.3 Hubungan 1:1 (one-to-one)**

- b. One-to-many (1:M). Relasi ini terjadi bila setiap instansi entitas dapat memiliki lebih dari satu hubungan terhadap instansi entitas lain tetapi tidak kebalikannya.



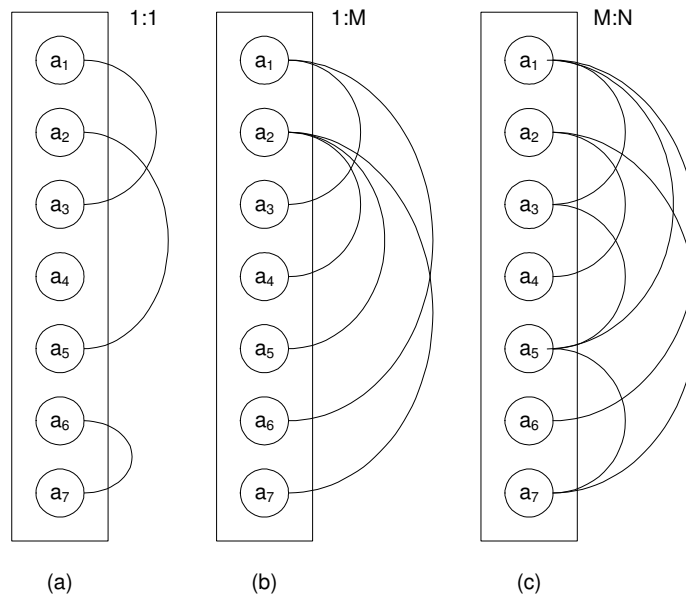
**Gambar 2.4 Hubungan 1:M (one-to-many)**

- c. Many-to-many (M:N). Hubungan saling memiliki lebih dari satu dari setiap instansi entitas terhadap instansi entitas lainnya.



**Gambar 2.5 Hubungan M:N (many-to-many)**

Selain relasi antara dua entitas, terdapat juga hubungan terhadap entitasnya sendiri yang disebut dengan *recursive relationship (self relation)*. Hubungan ini dapat mempunyai tipe biner, seperti yang sudah dibahas sebelumnya.



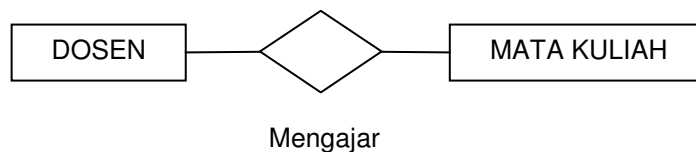
**Gambar 2.6 (a) Relasi rekursif 1:1 (b) Relasi rekursif 1:M (c) Relasi rekursif M:N**

**Partisipasi Hubungan**

Partisipasi atau keterlibatan setiap instansi entitas dalam membentuk hubungan dapat bersifat wajib (*obligatory*/digambarkan dengan garis penuh) atau tidak wajib (*non-obligatory*/digambarkan dengan garis putus-putus) dalam aturan data. Misalnya hubungan antara DOSEN dan Mata Kuliah dengan aturan data sebagai berikut:

*Setiap dosen harus mengajar satu mata kuliah dan setiap mata kuliah harus diajarkan oleh seorang dosen.*

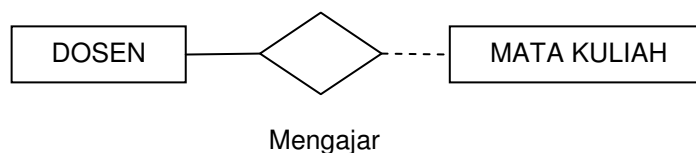
Setiap anggota entitas dalam hubungan adalah wajib untuk instansi entitas DOSEN maupun Mata Kuliah.



**Gambar 2.7 Diagram E-R dengan partisipasi wajib**

Dalam bentuk aturan lain yaitu:

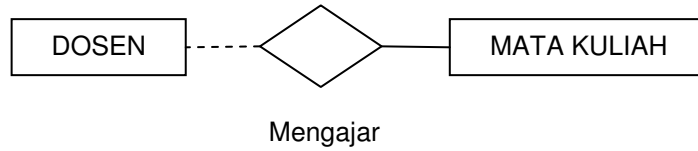
*Setiap dosen harus mengajar satu mata kuliah dan setiap mata kuliah mungkin diajarkan oleh seorang dosen atau mungkin tidak sama sekali.*



**Gambar 2.8 Diagram E-R dengan partisipasi wajib pada sisi Dosen dan tidak wajib pada sisi Mata Kuliah**

Atau aturan lain:

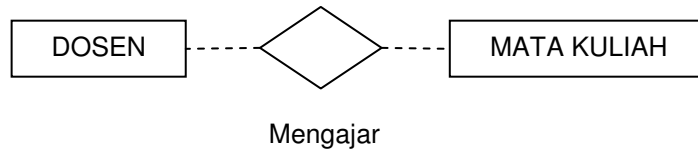
*Setiap dosen hanya boleh mengajar satu mata kuliah atau boleh saja dosen tidak mengajar dan setiap mata kuliah harus diajarkan oleh seorang dosen.*



**Gambar 2.9 Diagram E-R dengan partisipasi tidak wajib pada sisi Dosen dan wajib pada sisi Mata Kuliah**

Aturan lain yang tidak mewajibkan kedua belah pihak:

*Setiap dosen hanya boleh mengajar satu mata kuliah atau boleh saja dosen tidak mengajar dan setiap mata kuliah hanya boleh diajarkan oleh seorang dosen atau tidak sama sekali.*



**Gambar 2.10 Diagram E-R dengan partisipasi tidak wajib pada kedua sisi**



## PERANCANGAN BASIS DATA

Permasalahan dalam perancangan basis data adalah *bagaimana merancang struktur logikal dan fisik dari satu atau lebih basis data untuk memenuhi kebutuhan informasi yang diperlukan oleh pengguna sesuai dengan aplikasi-aplikasi yang ditentukan.* [Waliyanto2000]

Dengan permasalahan tersebut dapat ditentukan beberapa tujuan utama perancangan basis data, yaitu:

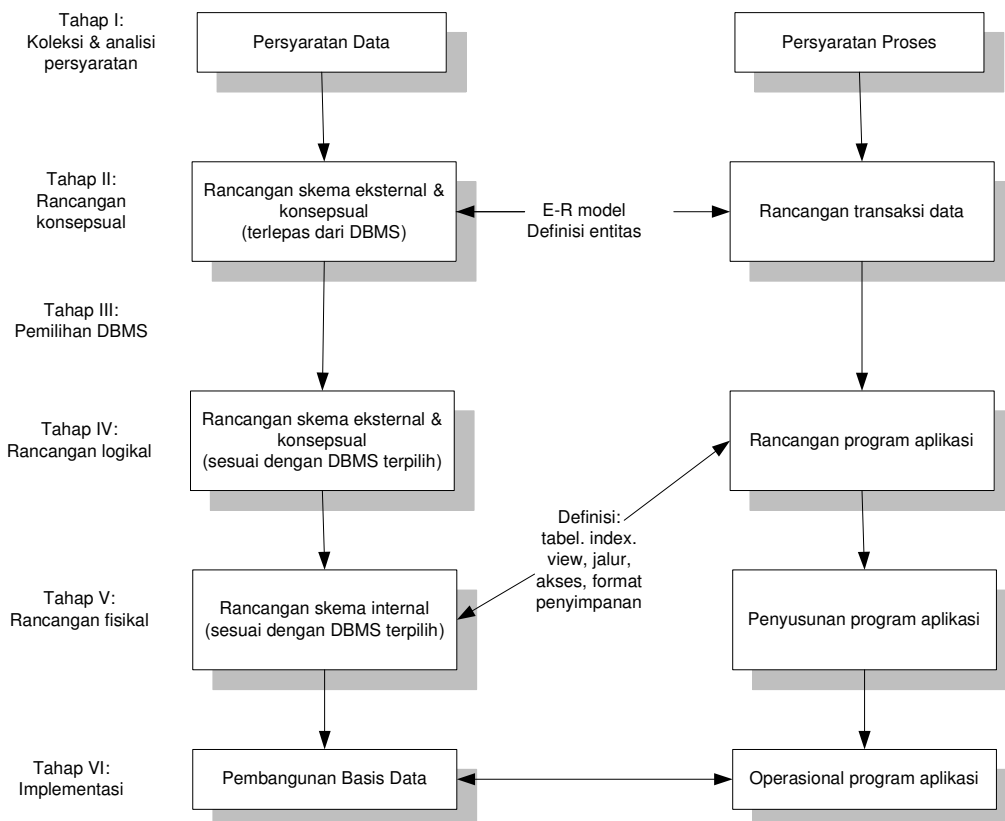
- Memenuhi kebutuhan informasi sesuai dengan yang diperlukan oleh pengguna untuk aplikasi tertentu.
- Mempermudah pemahaman terhadap struktur informasi yang tersedia dalam basis data,
- Memberikan keterangan tentang persyaratan pemrosesan dan kemampuan sistem, seperti lama tidaknya mengakses data, kapasitas memori yang tersedia dan sebagainya.

Tahapan-tahapan proses perancangan untuk memenuhi tujuan tersebut adalah:

- Mengumpulkan dan menganalisis persyaratan
- Merancang konseptual basis data
- Memilih Sistem Manajemen Basis Data
- Merancang logikal basis data
- Merancang fisik basis data (pemetaan model data)
- Implementasi sistem basis data

Dalam pelaksanaan perancangan tersebut terdapat dua kegiatan yang dapat dilakukan secara paralel, yaitu perancangan struktur dan isi data (analisis data) dan perancangan pemrosesan data serta program aplikasi (analisis fungsional).

Tahapan rancangan basis data seperti pada bagan di bawah ini tidak secara ketat harus diikuti secara berurutan. Karena antara tahap yang satu dengan yang lainnya dapat saling mempengaruhi dan memberi umpan balik.



Gambar 3.1 Tahapan perancangan basis data (kompilasi dari Elmasri R, 1994)

Rancangan konseptual basis data (tahap 2) menghasilkan skema konseptual dari basis data yang bebas dari DBMS tertentu. Dalam hal ini juga digunakan pemodelan bahasa tingkat tinggi seperti model E-R (*Entity Relationship*) atau EER (*Enhanced Entity Relationship*). Tahap ini juga menentukan transaksi data yang dapat dilakukan terhadap sistem basis data.

Rancangan logikal (tahap 4) disebut juga pemetaan model data, yaitu mentransformasikan model data yang telah dibuat pada tahap dua ke dalam model data yang sesuai dengan DBMS terpilih. Tahap ini juga melakukan perancangan skema eksternal untuk aplikasi yang ditentukan.

Rancangan fisik basis data (tahap 5) melakukan pendefinisian basis data yang akan disimpan sesuai dengan SMBD yang digunakan, meliputi struktur penyimpanan data, format data dan jalur akses. Tahap ini disebut skema internal.

### ***Koleksi dan Analisis Persyaratan***

Koleksi dan analisis persyaratan merupakan proses pengumpulan dan analisis tujuan dan harapan pengguna untuk memperoleh informasi dari sistem basis data. Kegiatan-kegiatan yang dilakukan dalam tahapan ini adalah sebagai berikut:

- a. Melakukan identifikasi bidang aplikasi dan kelompok pemakai
- b. Mempelajari dan menganalisis dokumen yang ada pada aplikasi tertentu
- c. Mempelajari sistem yang sedang berjalan
- d. Membuat semacam pertanyaan/angket pada calon pengguna yang dipandang potensial untuk memperoleh spesifikasi informasi dan proses yang diperlukan.

### ***Perancangan Konseptual Basis Data***

Tahapan ini meliputi dua kegiatan yaitu rancangan skema konseptual tentang organisasi data yang harus disimpan dalam basis data, dan rancangan transaksi yang dilakukan untuk memperoleh informasi dari sistem basis data hasil analisis persyaratan pada tahap 1.

#### **Rancangan skema konseptual**

Hasil rancangan konseptual merupakan pemodelan data dari pemahaman dunia nyata yang dituliskan dalam bahasa tingkat tinggi dan tidak terikat dengan DBMS yang akan digunakan. Umumnya pembuatan skema konseptual ini menggunakan diagram E-R.

Untuk menyusun rancangan konseptual dimulai dengan identifikasi komponen utama dari skema (entitas, hubungan, atribut) dengan mengacu pada karakteristik sebagai berikut:

- a. Model data harus cukup memberikan tampilan yang menggambarkan perbedaan jenis data, hubungan dan *constraint* (ekspresif).
- b. Model harus dibuat sederhana dan mudah dipahami serta digunakan oleh pengguna.
- c. Penyajian model data dibuat dalam diagram yang mudah diinterpretasi
- d. Penyajian model data dalam skema harus teliti dan tidak menimbulkan interpretasi (akurat).

#### **Rancangan transaksi**

Teknik pembuatan spesifikasi transaksi dilakukan dengan melakukan identifikasi data masukan dan data keluaran serta sifat fungsional transaksi, sehingga perancang dapat membuat model konseptual transaksi yang tidak terikat dengan sistem.

Fungsi-sungsi model transaksi adalah sebagai berikut:

- a. Transaksi pemanggilan (*retrieval transaction*), yaitu pemanggilan data untuk ditampilkan di layar monitor atau dicetak sebagai laporan.
- b. Transaksi pembaharuan (*update transaction*), digunakan untuk pemasukan data baru atau perubahan data lama.
- c. Transaksi campuran (*mixed transaction*), digunakan untuk kombinasi pemanggilan data dan pembaharuan data.

### ***Pemilihan DBMS***

Faktor-faktor yang menentukan pemilihan DBMS antara lain adalah faktor teknik, faktor ekonomi dan politik dalam organisasi.

Faktor teknik meliputi kelangsungan dari DBMS untuk diterapkan dalam pengelolaan data seperti jenis model DBMS, struktur penyimpanan data dan alur akses data, interface pengguna dan pemrogram, jenis bahasa tingkat tinggi dan sebagainya.

Faktor ekonomi diantaranya pembelian software DBMS, pembelian hardware, biaya pemeliharaan sistem, biaya penyusunan basis data dan lain sebagainya.

### ***Pemodelan Logikal Basis Data***

Tujuan dari tahap ini adalah menyusun rancangan konseptual dan skema eksternal yang sesuai dengan DBMS yang dipilih. Langkah-langkah yang dilakukannya adalah:

- a. Pemetaan (transformasi data) yang tidak terikat sistem.
- b. Penyusunan skema sesuai dengan DBMS.

### ***Perancangan Fisikal Basis Data***

Tujuan dari perancangan ini adalah untuk membuat spesifikasi struktur penyimpanan dan jalur akses data sehingga diperoleh kemampuan sistem yang baik untuk berbagi aplikasi. Beberapa hal yang menjadi pertimbangan dalam perancangan fisikal adalah:

1. Waktu tanggap. Yaitu waktu yang digunakan oleh sistem sejak transaksi basis data dimasukkan untuk dieksekusi sampai mendapat tanggapan dari sistem. Faktor yang mempengaruhinya adalah waktu akses basis data yang dikendalikan oleh DBMS serta dipengaruhi oleh sistem pemuatan data (*loading*) pada komputer, sistem operasi yang digunakan, atau penundaan sistem komunikasi.
2. Penggunaan memori komputer. Merupakan kapasitas memori komputer yang digunakan untuk menyimpan berkas-berkas basis data dan struktur jalur akses.
3. Transaksi data. Kemampuan melakukan transaksi data tiap satuan waktu merupakan hal yang kritis.

### ***Implementasi Sistem Basis Data***

Tahap ini merupakan implementasi dari hasil pemodelan logikal dan fisikal. Bahasa yang digunakan untuk definisi data atau penyimpanan data yang sesuai dengan DBMS terpilih. Implementasi penyusunan basis data dimulai dari pembuatan berkas-berkas data kosong yang akan digunakan untuk menyimpan data dalam basis data. Kemudian dilanjutkan dengan pemasukan data untuk tiap instansi tabel.

Dalam implemmentasi rancangan transaksi, program aplikasi ditulis dengan bahasa manipulasi data yang sesuai. Program-program aplikasi yang dibuat harus dilakukan uji coba dulu untuk menguji kebenaran program. Setelah diuji kemudian diimplementasikan dalam operasional sistem basis data.



Referensi:

- [Abdul1999] Abdul Kadir. 1999. ***Konsep & Tuntunan Praktis Basis Data***. Penerbit Andi. Yogyakarta.
- [David2002] David M. Kroenke. 2002. ***Database Processing Fundamentals, Design, and Implementation***. Eight Edition. Prentice Hall.
- [Ramez2000] Ramez Elmasri & Shamkant B Navathe. 2000. ***Database System***.
- [R.E. 2003] R.E. Wyllys. 2003. ***Database-Management Principles And Applications***.
- [Sitansu1991] Sitansu S. Mittra. 1991. ***Principles of Relational Database Systems***. International Editions. Prentice-Hall. New Jersey.
- [Waliyanto2000] Waliyanto. 2000. ***Sistem Basis Data Analisis dan Pemodelan Data***. J&J Learning. Yogyakarta.