

1. Pertemuan 1

1.1 Sistem perkuliahan

- a. Perkuliahan diselenggarakan 14 kali pertemuan (2 SKS)
- b. Perkuliahan diselenggarakan 14 kali pertemuan (2 SKS)
- c. Wajib kehadiran Mahasiswa 80% (-3 kali tidak masuk)
- d. Materi perkuliahan akan diberikan salinannya kepada Mahasiswa
- e. Mahasiswa dianjurkan membawa flashdisk
- f. Batas keterlambatan 15 menit setelah perkuliahan dimulai
- g. Mahasiswa diperbolehkan berkonsultasi dengan dosen; mengenai materi perkuliahan secara personal atau kelompok di luar jam perkuliahan (tatap muka; via email; kuliah online)
- h. Mengikuti tata tertib Lab
- i. Tidak diperbolehkan menggunakan perangkat komunikasi selama perkuliahan (setting silent/vibrate)
- j. Bersikap sopan dan tidak mengganggu keberlangsungan perkuliahan
- k. Tersedia waktu Shalat bagi yang beragama Islam.

1.2 Tujuan dan cakupan materi perkuliahan

Cakupan atau ruang lingkup materi perlu ditentukan untuk mengetahui apakah materi yang akan diajarkan terlalu banyak, terlalu sedikit, atau telah memadai sehingga terjadi kesesuaian dengan kompetensi dasar yang ingin dicapai. Panduan Pengembangan Materi Pembelajaran

1.3 Silabus

Minggu ke 1 : Pengantar perkuliahan

Sistem Perkuliahan| Tujuan dan cakupan materi perkuliahan| Silabus| Pemberitahuan daftar pustaka| Tinjauan ulang mata kuliah prasyarat.

Minggu ke 2 : Sistem Basis Data

Pengenalan basis data| komponen system basis data| Abstraksi basis data| Bahasa basis data| Struktur sistem basis data| Siklus hidup perancangan basis data.

Minggu ke 3 : Basis Data Relasional

Definisi basis data relasional| Elemen basis data relasional| Istilah-istilah basis data relasional| Sifat relasi.

Minggu ke 4 : Normalisasi Data

Pengantar normalisasi| Anomali| Depedensi| Bentuk normal.

Minggu ke 5 : Praktikum 1: Normalisasi Data

Analisis sistem informasi dan menyelesaikan sebuah kasus normalisasi.

Minggu ke 6 : Pemodelan Data

Membuat Model Entity-Relationship (E-R) 2. Varian entitas| Varian atribut| Varian relasi| Spesialisasi dan generalisasi| Agregasi| Asosiatif| Tahapan pembuatan diagram Entity-Relationship (E-R)| Masalah pada model Entity-Relationship (E-R).

Minggu ke 7 : Praktikum 2: Pemodelan Data

Normalisasi data hasil praktikum 1| Pembangunan diagram Entity-Relationship (E-R).

Minggu ke 8 : Ujian Tengah Semester

Minggu ke 9: Penerapan Basis Data

Transformasi diagram Entity-Relationship (E-R) ke basis data fisik| Relasi table| Struktur file | Kodifikasi| Aplikasi basis data.

Minggu ke 10: Denormalisasi Basis Data

Redundansi dan bentuk-bentuk denormalisasi| Atribut turunan| Atribut yang berlebihan| Tabel rekapitulasi.

Minggu ke 11: Bahasa Query

Data Definition Language (DDL)| Data Manipulation Language (DML).

Minggu ke 12 : Pengamanan Basis Data

Pemberian otoritas| Perintah SQL untuk pengamanan basis data| Penyandian (Enkripsi)| Pemeliharaan integritas basis data .

Minggu ke 13 : Persentasi Tugas Kelompok

Pengumpulan laporan perancangan basis data| Persentasi hasil perancangan basis data| Tanya jawab.

Minggu ke 14 : Persentasi Tugas Kelompok

Pengumpulan laporan perancangan basis data| Persentasi hasil perancangan basis data| Tanya jawab.

Minggu ke 15 : Persentasi Tugas Kelompok

Pengumpulan laporan perancangan basis data| Persentasi hasil perancangan basis data| Tanya jawab.

Minggu ke 16 : Ujian Akhir Semester

2. System Basis data

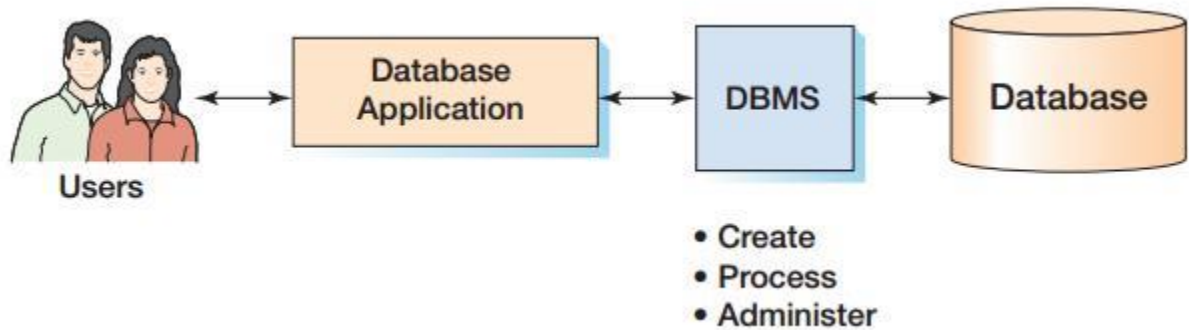
2.1. Pengenalan Basis data

Terdiri dari 2 kata, yaitu :

- Basis
Dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul
- Data
 - adalah representasi fakta dunia nyata yang mewakili suatu objek
 - Objek terbagi menjadi 2, yaitu:
 - Objek Nyata
 - Objek Abstrak (tidak nyata)
 - yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya

Basis data adalah suatu kumpulan data terhubung (interrelated data) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data(controlled redundancy) dengan cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali; dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang menggunakannya; data disimpan sedemikian rupa sehingga penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

2.2. komponen-komponen basisdata



1. **Basis Data** sebagai inti dari sistem basis data Yakni basis data yang mewakili sistem tertentu untuk dikelola. Sebuah sistem basis data bisa terdiri dari lebih dari satu basis data.
2. **Perangkat Lunak atau Software sistem operasi (Operating System)** untuk mengelola basis data merupakan perangkat lunak yang bersifat opsional untuk memfungsikan, mengendalikan seluruh sumber daya, serta melakukan operasi dasar pada sistem komputer. Contoh *software* OS yakni Windows 9x, Linux, Unix, Windows 2000/XP). *Software* OS harus kompatibel atau sesuai dengan *software* pengelola basis data yang digunakan.
3. **Hardware atau perangkat keras** sebagai pendukung operasi pengolahan data biasanya berupa perangkat komputer standar, media penyimpan sekunder dan media komunikasi untuk sistem jaringan.
4. **Manusia atau pengguna (user)** sebagai perancang, pengelola atau pemakai Orang-orang yang berinteraksi dengan sistem basis data, mulai dari yang merancang sampai yang menggunakan di tingkat akhir.
 - **Database Administrator**, yaitu *user* yang menjadi pusat pengendali seluruh sistem *database*. Berwenang untuk menentukan pola struktur *database*, memodifikasinya, membagi tugas pengolahan, dan sebagainya.
 - **Application Programmer**, yaitu *user* yang membuat program aplikasi untuk berinteraksi dengan sistem atau mengakses *database* dengan cara pemanggilan bahasa pemrograman tertentu.

- **Sophisticated User**, yaitu *user* yang melakukan *request* (permintaan) untuk berinteraksi dengan sistem dalam bentuk bahasa *query database*, seperti menggunakan MS-Access, SQL, dan sebagainya.
- **Specialized User**, yaitu user yang menulis aplikasi *database* tidak untuk pemrosesan data secara tradisional melainkan untuk keperluan khusus seperti pengolahan citra, sistem pakar, atau aplikasi *artificial intelegent*.
- **Naive User**, yaitu *user* umum yang tidak memerlukan keahlian khusus dalam pemrograman dan cukup menjalankan aplikasi yang tersedia untuk berinteraksi dengan sistem untuk mengakses data.

5. Database Management System (DBMS)

DBMS (databases management system) dapat diartikan sebagai suatu program komputer yang dapat kita gunakan untuk memasukkan, mengubah, menghapus, memodifikasi serta memperoleh data/informasi dengan praktis dan efisien. Dengan kata lain DBMS merupakan suatu perangkat lunak yang dapat digunakan untuk mengelola dan memanggil (*query*) pada basis data.

KELEBIHAN DBMS

Kelebihan dari DBMS antara lain adalah:

- Kepraktisan.
Maksud dari kepraktisan yaitu DBMS menyediakan media penyimpanan yang permanen yang berukuran kecil namun dapat banyak menyimpan data-data jika dibandingkan dengan menggunakan kertas.
- Kecepatan.
Maksud dari kecepatan atau cepat yaitu dengan adanya DBMS Komputer dapat mencari dan menampilkan informasi yang kita dibutuhkan dengan cepat.

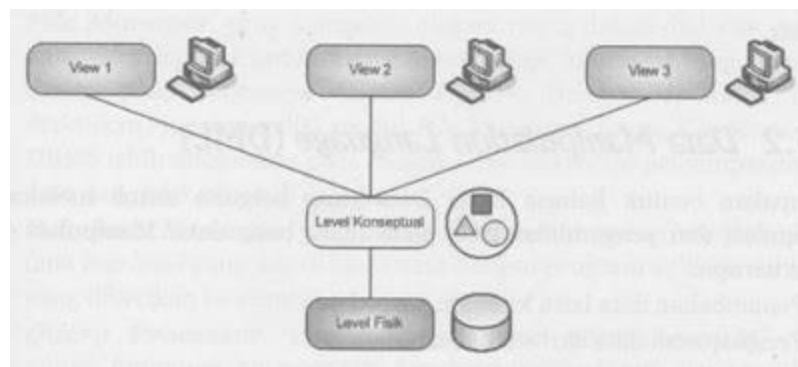
- Mengurangi kejemuhan.

Maksud dari mengurangi kejemuhan yaitu ketika melakukan pekerjaan, Pekerjaan yang berulang-ulang dapat menimbulkan kebosanan bagi manusia, sedangkan mesin tidak merasakannya.

- Update to date.

Maksud dari update to date yaitu Informasi yang tersedia selalu berubah dan akurat setiap

2.3. Abstraksi data



Salah satu tujuan dan DBMS adalah untuk menyediakan antarmuka (interface) dalam mengelola data yang lebih ramah (user friendly) kepada pemakai. Untuk itu, sistem tersebut akan menyembunyikan detail tentang bagaimana data disimpan dan dikelola. Karena itu, seringkali data yang terlihat oleh seorang pemakai dapat berbeda dengan yang sesungguhnya tersimpan secara fisik. Abstraksi data mengacu pada tingkatan/level dalam bagaimana melihat data dalam sebuah sistem basis data.

Ada 3 level abstraksi data:

1. Level Fisik (Physical Level)

Merupakan level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada level ini, pemakai melihat data sebagai gabungan dan struktur dan datanya sendiri.

Pemakai juga berkompeten dalam mengetahui bagaimana representasi fisik dan penyimpanan data. Pada level ini kita berurusan dengan data sebagai teks, sebagai angka, atau bahkan melihatnya sebagai himpunan bit data.

2. **Level Logik/Konseptual (Conceptual Level)**

Merupakan level berikutnya dalam abstraksi data yang menggambarkan data secara fungsional yang disimpan dalam basis data serta hubungannya dengan data yang lain. Pemakai pada level ini, misalnya,

mengetahui bahwa data pegawai disimpan dalam beberapa tabel, seperti tabel pribadi, tabel pendidikan, tabel pekerjaan, tabel keluarga, dan sebagainya.

3. **Level Penampakan (View Level)**

Merupakan level tertinggi dan abstraksi data yang hanya menunjukkan sebagian dan basis data. Banyak user dalam sistem basis data tidak akan terlibat dengan semua data/informasi yang disimpan. Para user.

umumnya hanya membutuhkan sebagian data dalam basis data yang kemunculannya di mata pemakai diatur oleh aplikasi end-user. Aplikasi ini juga yang mengonversi data ash (fisik) menjadi data bermakna (logik) pada pemakai. Misalnya, data hari yang disimpan dalam bentuk kode (1 untuk Senin, 2 untuk Selasa, dan seterusnya) yang kemudian ditampilkan bukan dalam bentuk kodenya (1,2,3, dan seterusnya) tapi sudah dalam bentuk nama harinya (Senin, Selasa, Rabu, dan seterusnya). Data yang 'dinikmati' pemakai juga bahkan sama sekali berbeda dengan representasi fisiknya, misalnya untuk data yang dapat divisualkan sebagai gambar, data yang dapat diperdengarkan sebagai suara, dan sebagainya. Data yang diperlihatkan juga bisa saja tidak berasal dan hanya sebuah tabel tapi mewakili relasi antartabel, tapi bagi pemakai yang menggunakannya terasa sebagai satu kesatuan data yang kompak.

2.4. Bahasa basis data

2.4.1. Deskripsi Bahasa Basisdata

Bahasa basis data umumnya dapat ditempelkan (embedded) ke bahasa pemrograman 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*).

2.4.2. Komponen Bahasa Basisdata

1. *Data Definition Language* (DDL)

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

2. *Data Manipulation Language* (DML)

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 operasi *query*. Sebenarnya DML tidak hanya berisi operasi untuk *query*, namun juga meliputi operasi penghapusan, pembaruan dan penyisipan.

3. *Data Control Language* (DCL)

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

2.4.3. 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 (column / field). Di setiap pertemuan baris data dan kolom itulah item-item data (satuan data terkecil) ditempatkan.

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

nid	nama_d	tempat_lahir	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 Pujliningsih,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.7	Bekasi Timur	45612	1150000
01001	Cokro Dinatingrat,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

2.5. Siklus hidup perancangan basis data

Siklus hidup aplikasi basis data berhubungan dengan siklus hidup sistem informasi. Siklus kehidupan sistem informasi sering disebut macro life cycle, dimana siklus kehidupan basis data merupakan micro life cycle. Proses perancangan basis data merupakan bagian dari siklus hidup sistem informasi.

Ada 6 fase proses perancangan basis data, yakni:

1. Pengumpulan data dan analisa

Proses identifikasi dan analisa kebutuhan-kebutuhan data disebut pengumpulan data dan analisa. Untuk menentukan kebutuhan-kebutuhan suatu sistem basis data, pertama harus mengenal bagian-bagian lain dari sistem informasi yang akan berinteraksi dengan sistem basis data, termasuk para

pemakai yang ada dan para pemakai yang baru serta aplikasi-aplikasinya. Kebutuhan-kebutuhan dari para pemakai dan aplikasi inilah yang kemudian dikumpulkan dan dianalisa.

Ada 4 aktivitas pengumpulan data dan analisis, yaitu:

- Menentukan kelompok pemakai dan bidang-bidang aplikasinya.
- Peninjauan dokumentasi yang ada.
- Analisa lingkungan operasi dan pemrosesan data.
- Daftar pertanyaan dan wawancara

2. Perancangan basis data secara konseptual

Tujuan dari fase ini adalah menghasilkan conceptual schema untuk basis data yang tergantung pada sebuah DBMS yang spesifik. Sering menggunakan sebuah high-level data model seperti ERD (Entity Relationship Diagram) model selama fase ini. Dalam conceptual schema, kita harus memerinci aplikasi-aplikasi basis data yang diketahui dan transaksi-transaksi yang mungkin.

Fase perancangan basis data secara konseptual mempunyai 2 aktifitas paralel :

- Perancangan skema konseptual :

Berfungsi untuk menguji kebutuhan-kebutuhan data dari suatu basis data yang merupakan hasil dari fase 1, dan menghasilkan sebuah conceptual basis data schema pada DBMS independent model data tingkat tinggi seperti ERD (Entity Relationship Diagram) model. Skema ini dapat dihasilkan dengan menggabungkan bermacam-macam kebutuhan pengguna dan secara langsung membuat skema basis data atau dengan merancang skema-skema yang terpisah dari kebutuhan tiap-tiap pengguna dan kemudian menggabungkan skema-skema tsb. Model data yang digunakan pada perancangan skema konseptual adalah DBMS-independent, dan langkah selanjutnya adalah memilih sebuah DBMS untuk melaksanakan rancangan tsb.

- Perancangan transaksi :

Berfungsi untuk menguji aplikasi-aplikasi basis data dimana kebutuhan-kebutuhannya telah dianalisa pada fase 1, dan menghasilkan perincian transaksi-transaksi ini. Pada tahap ini merupakan pembuatan

flowchart dan kegunaan fase ini yang diproses secara paralel bersama fase perancangan skema konseptual adalah untuk merancang karakteristik dari transaksi-transaksi basis data yang telah diketahui pada suatu DBMS-independent. Transaksi-transaksi ini akan digunakan untuk memproses dan memanipulasi basis data suatu saat dimana basis data tsb dilaksanakan.

3. Pemilihan DBMS

Pemilihan basis data ditentukan oleh beberapa faktor, diantaranya : faktor teknik, ekonomi dan organisasi.

Contoh faktor teknik :

Keberadaan DBMS dalam menjalankan tugasnya seperti jenis-jenis DBMS (relational, network, hierarchical), struktur penyimpanan, dan jalur akses yang mendukung DBMS, pemakai, dll.

4. Perancangan basis data secara fisik

Perancangan basis data secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada file-file basis data untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi. Selama fase ini, dirancang spesifikasi-spesifikasi untuk basis data yang disimpan yang berhubungan dengan struktur-struktur penyimpanan fisik, penempatan record dan jalur akses. Berhubungan dengan internal schema (pada istilah 3 level arsitektur DBMS). Pada tahap ini, perancangan fisik telah ditujukan untuk system DBMS tertentu. Perancangan basis data tingkat fisik sudah dikaitkan dengan platform dan perangkat lunak system manajemen basis data dimana basis data diimplementasikan.

5. Implementasi sistem basis data

Setelah perancangan secara logika dan secara fisik lengkap, kita dapat melaksanakan sistem basis data. Perintah-perintah dalam DDL dan DML (Data Manipulation Language) dari DBMS yang dipilih, dihimpun dan digunakan untuk membuat skema basis data dan file-file basis data (yang kosong). Sekarang basis data tsb dimuat (disatukan) dengan datanya. Jika data harus dirubah dari

sistem komputer sebelumnya, perubahan-perubahan yang rutin mungkin diperlukan untuk format ulang datanya yang kemudian dimasukkan ke basis data yang baru. Transaksi-transaksi basis data sekarang harus dilaksanakan oleh para programmer aplikasi.

6. Basis Data Relasional

6.1 Definisi basis data relasional

Model Data Relasional adalah model data yang ditemukan oleh E.F. Codd berdasarkan teori relasional seperti aljabar dan kalkulus relasional. Model data ini menggunakan tabel berdimensi dua (sering disebut dengan relasi / table) untuk menggambarkan sebuah berkas data dan menggunakan kunci tamu (foreign key) untuk menghubungkan suatu tabel dengan tabel yang lain.

Contoh table dan keterhubungannya

MHS

NPM	Nama	Alamat
10156621	Nurhayati	Bekasi
10177581	Rosalin	Bekasi
20119877	Beni	Depok
35528899	Prameswari	Bogor
43221234	Pipit	Jakarta
53341150	Wira	Jakarta

Nilai_akademik

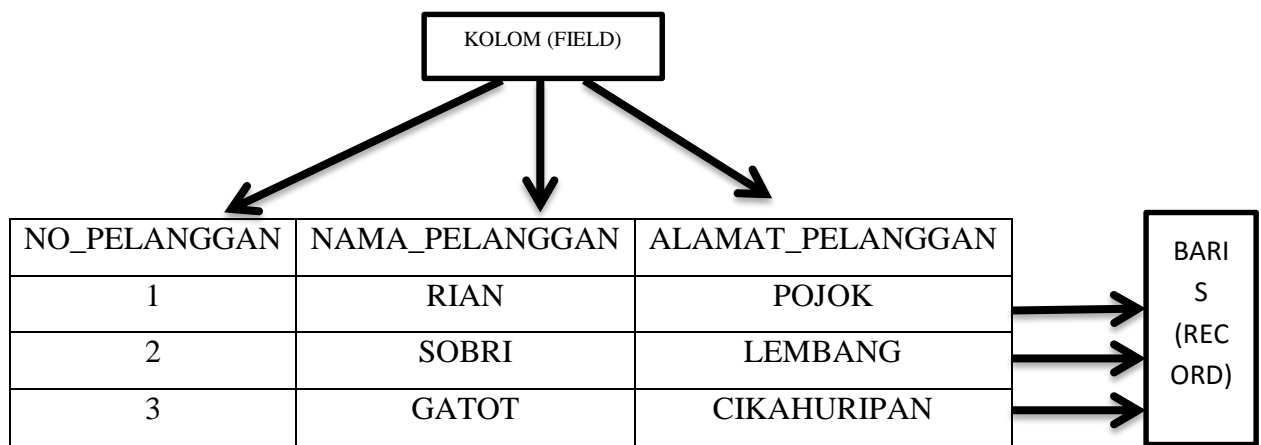
NPM	KDMK	MID	FINAL
10156621	KD123	70	80
10177581	KK234	75	85
20119877	KK234	65	70
35528899	KU321	90	85
43221234	KD123	80	85
53341150	KU321	65	90
53341150	KK234	75	95

6.2 Elemen basis data relasional

Elemen-elemen basis data relasional terdiri dari:

1. Tabel

Tabel merupakan kumpulan informasi secara logis yang terkait dan diperlakukan sebagai unit. Setiap tabel selalu terdiri atas lajur vertikal yang biasa disebut dengan kolom atribut (column/field) dan lajur horizontal yang biasa disebut dengan baris data (row/record). Di setiap pertemuan kolom atribut dan baris data ditempatkan item-item data (satuan data terkecil).



2. Kolom (field)

Kolom merupakan atribut data yang ada pada suatu tabel. Pada gambar 1 tabel pelanggan memiliki 3 atribut data (kolom) diantaranya no_pelanggan, nama_pelanggan, almt_pelanggan.

3. Baris (record)

Suatu tabel terdiri dari baris (record) yang mengisi setiap kolom (field). Baris adalah kejadian tunggal yang berisi data di dalam tabel. Setiap baris diperlakukan sebagai unit tunggal. Pada gambar 1 tabel pelanggan terdapat 3 baris berisi informasi setiap individu pelanggan yang dijelaskan oleh kolom di atasnya.

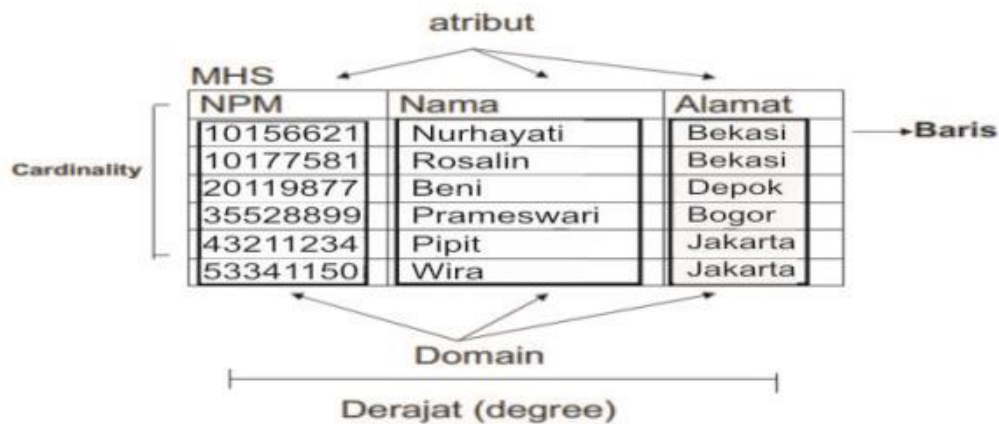
4. Kunci

Pada umumnya terdapat dua jenis kunci diantaranya kunci utama (primary key) dan kunci tamu (foreign key). Kunci utama adalah suatu kolom yang memiliki nilai unik dan digunakan untuk mengidentifikasi setiap baris di dalam tabel. Karena sifatnya yang unik maka dapat digunakan sebagai acuan dalam menampilkan, mengubah dan menghapus baris data. Dengan adanya kunci utama ini juga dapat mencegah terjadinya duplikasi data karena sifatnya yang unik. Kunci utama memiliki karakteristik sebagai berikut:

- Mandatory; bahwa kolom yang menjadi kunci utama tidak boleh menyimpan nilai null karena dapat menyebabkan duplikasi baris data.
- Unique; nilai dari kunci utama bersifat unik atau tidak ada kesamaan dengan nilai lainnya. Seperti contoh pada gambar 1 tabel pelanggan kolom no_pelanggan merupakan kunci utama karena nilainya yang unik (tidak sama dengan yang lainnya) sedangkan nama_pelanggan tidak bisa dijadikan sebagai kunci utama karena ada kemungkinan nama yang sama untuk setiap pelanggan yang berbeda.
- Stable; nilai dari kunci utama bersifat stabil atau tidak berubah-ubah. Seperti contoh pada gambar 1 tabel pelanggan kolom no_pelanggan merupakan kunci utama karena nilainya stabil (tidak berubah-ubah) sedangkan alamat_pelanggan memungkinkan nilainya berubah apabila pelanggan yang bersangkutan pindah rumah.

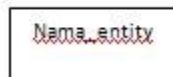
- Short; memiliki sedikit karakter, karena dapat berpengaruh pada ruang penyimpanan yang sedikit, pencarian data menjadi lebih cepat, dan meminimalisir dalam kesalahan pemanggilan serta memasukan data. Seperti contoh pada gambar 1 tabel pelanggan kolom no_pelanggan memiliki jumlah karakter yang lebih sedikit ketimbang dengan alamat_pelanggan

6.3 Istilah-istilah basis data relasional



1. Entitas (Entity) :

merupakan suatu objek yang dapat dibedakan dari yang lainnya dan dapat diwujudkan dalam basis data. Objek tersebut dapat berupa orang, benda, ataupun peristiwa. Contoh entitas dalam lingkungan universitas, terdiri dari : mahasiswa, mata kuliah dsb. Kumpulan dari entitas disebut himpunan entitas, contoh : semua mahasiswa.



2. Relasi (Relation) :

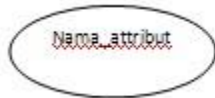
merupakan suatu tabel yang terdiri dari beberapa kolom dan baris. Relasi menunjukkan adanya korelasi di antara sejumlah entitas yang asalnya dari himpunan entitas yang berbeda.



Contoh : tabel nilai mahasiswa, tabel mata kuliah dsb.

3. Atribut (Attribute) :

merupakan kolom yang terdapat dalam sebuah relasi (field). Atribut mendeskripsikan setiap karakter yang menjadi ciri suatu entitas.



4. Tuple : merupakan baris yang terdapat dalam sebuah relasi (record) atau kumpulan elemen-elemen yang saling terkait menginformasikan suatu entitas secara lengkap.

5. Domain : merupakan kumpulan nilai yang valid dari satu atau lebih atribut (field).

6. Derajat (Degree) : merupakan banyaknya atribut / kolom yang terdapat dalam suatu relasi (tabel).

7. Kardinalitas (Cardinality) : merupakan banyaknya tuple / baris yang terdapat dalam sebuah relasi (tabel).

8. Kunci Kandidat

Kunci kandidat adalah sebuah atribut atau gabungan beberapa atribut yang digunakan untuk membedakan antara satu tuple dengan tuple lainnya. Dengan kata lain kunci tersebut dapat bertindak sebagai identitas yang unik bagi baris-baris dalam suatu relasi.

9. Kunci Primer/Utama

Kunci primer atau kunci utama adalah kunci kandidat yang dipilih sebagai identitas untuk membedakan satu tuple dengan tuple lain dalam suatu relasi. Perlu diketahui dalam basis data relasional, sebuah relasi harus memiliki satu kunci primer saja

10. Kunci Asing/Tamu

Kunci asing atau kunci tamu adalah sebuah atribut atau gabungan dari beberapa atribut dalam suatu relasi yang merujuk (merefrensi) ke kunci primer relasi lain. Kunci asing dalam suatu relasi yang mengacu pada kunci primer milik relasi lain merupakan perwujudan untuk membentuk hubungan antar relasi.

6.4 Sifat relasi

Relasi disini diartikan sebagai sebuah tabel yang memiliki 2 dimensi yang terdiri atas sejumlah kolom/atribut dan sejumlah baris data. Relasi pada basis data relasional memiliki sifat-sifat sebagai berikut:

1. Setiap relasi dalam basis data harus memiliki nama yang unik (tidak kembar).
2. Setiap sel (perpotongan antara baris dan kolom) dalam relasi harus bersifat atomic (bernilai tunggal).
3. Setiap nama kolom/atribut dalam tabel harus memiliki nama yang unik (tidak kembar).
4. Nilai untuk setiap atribut harus berdomain sama.
5. Setiap baris data harus memiliki kunci utama yang dapat dibedakan atau bersifat unik (tidak kembar).

7. Normalisasi Data

7.1 Pengantar normalisasi

Normalisasi

- Kroenke mendefinisikan normalisasi sbagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tida memiliki masalah tersebut.
- Masalah yang dimaksud olej kroenke ini sering disebut dengan istilah anomali.

a) Definisi normalisasi

merupakan suatu pendekatan sistematis untuk meminimalkan redundansi data pada suatu database agar database tersebut dapat bekerja dengan optimal.

b) Jenis atribut

Pengertian Atribut (Field/Kolom)

Atribut identik dengan istilah “kolom data” tetapi dapat menunjukkan fungsinya sebagai pembentuk karakteristik (sifat-sifat) yang melekat dalam sebuah tabel.Pada

penerapan aturan normalisasi, bisa berdampak pada penghilangan atau penambahan kolom tertentu, atau bahkan dapat membentuk suatu tabel baru.

Selain penamaan yang unik berdasarkan fungsinya di tiap tabel, atribut juga dapat dibedakan berdasarkan sejumlah pengelompokan sbb :

a. Atribut Kunci

Atribut Kunci adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik.

1. Kunci Calon (Candidat key)

Kunci calon adalah salah satu rangkaian yang mempunyai nilai unik untuk membedakan atau mengidentifikasi nilai-nilai kombinasi yang unik diantara semua kejadian yang spesifik dari entetitas. Kunci calon ini tidak boleh berisi atribut dari tabel yang lain.

Contoh: table_pegawai

1. nip
2. no_ktp
3. nama
4. tempat_lahir
5. tanggal_lahir
6. alamat
7. Kota

Kunci Calon disini adalah:

- nip
- no_ktp

2. Kunci Utama (Primary Key)

Kunci utama adalah atribut merupakan kunci calon yang telah dipilih untuk mengidentifikasi setiap record secara unik. **Kunci utama** harus

merupakan atribut yang benar-benar unik dan tidak boleh ada nilai NULL. Kunci utama adalah suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam table. Salah satu dari kunci calon dapat dipilih menjadi kunci utama dengan 3 kriteria sbb:

- Kunci tersebut lebih natural untuk dijadikan acuan
- Kunci tersebut lebih sederhana
- Kunci tersebut cukup unik

No induk dan no ktp adalah kunci calon (Candidate Key) dan untuk kunci utama (primary key) adalah salah satu yang dipilih dari kunci calon. Misalnya No. induk di jadikan primary key, maka primary key nya adalah no induk.

3. Kunci Alternatif (Alternate Key)

Kunci Alternatif adalah kunci alternatif yang tidak terpilih. Misal : dalam suatu entitas terdapat dua atribut yang bisa dijadikan sebagai kunci. Sementara yang boleh dijadikan kunci hanya satu, maka anda harus memilih salah satu. Atribut yang dipilih, disebut **kunci utama**. sedangkan atribut yang tidak dipilih disebut dengan kunci .

Contoh:

Tabel **pegawai** berisi atribut

- nip
- no_ktp
- nama
- tempat_lahir
- tanggal_lahir
- alamat
- kota

nip dan **no_ktp** adalah kunci calon dan untuk kunci utama adalah salah satu yang dipilih dari kunci calon. Misalnya nip di jadikan kunci utama, maka no_ktp otomatis menjadi kunci alternatif.

4. Kunci Tamu (Foreign Key)

Jika sebuah kunci utama terhubungan ke tabel lain, maka keberadaan kunci utama pada tersebut di sebut sebagai kunci tamu. Kunci tamu adalah Sebuah kumpulan atribut dalam satu relasi yang digunakan untuk me“refer” (menunjuk) ke suatu baris (tuple) pada relasi yang lain (harus berkorespondensi dengan kunci utama pada relasi yang kedua), seperti: ‘logical pointer’.

5. Kunci Komposit (Composite key)

Dalam desain basisdata, kunci komposit adalah kunci yang terdiri dari 2 atau lebih atribut yang secara unik mengidentifikasi suatu kejadian entitas. Setiap atribut yang membentuk kunci senyawa adalah kunci sederhana dalam haknya sendiri.

6. Kunci Sekunder (Secondary Key)

Kunci sekunder adalah sebuah atribut atau kombinasi yang digunakan hanya untuk tujuan pengambilan data.

b. Atribut Deskriptif

Atribut Deskriptif adalah atribut-atribut yang tidak menjadi atau merupakan anggota dari primary key. Jadi, dalam tabel mahasiswa yang menjadi atribut deskriptif adalah selain NIM.

c. Atribut Sederhana (simple attribute)

atribut Sederhana adalah atribut atomik yang tidak dapat dipilah lagi. Contoh Atribut Sederhana pada tabel **customer** adalah no_identitas dan jaminan, dimana atribut ini tidak bisa dipecah lagi.

d. Atribut Komposit (composite attribute)

Atribut Komposit (composite attribute) adalah atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna. Contoh pada tabel customer adalah atribut alamat, dimana dapat diuraikan lagi menjadi alamat, kota dan kode_pos.

e. Atribut Bernilai Tunggal (single-valued attribute)

Atribut bernilai tunggal adalah atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data. Contoh : Bila seorang mahasiswa memiliki 2 tempat tinggal, maka hanya 1 saja yang boleh diisikan ke atribut alamat_mhs.

f. Atribut Bernilai Banyak (multiple-valued attribute)

Atribut bernilai banyak adalah atribut-atribut yang dapat diisi dengan lebih dari satu nilai, tetapi jenisnya sama. Contoh : Atribut hobi pada data mahasiswa. Ada mahasiswa yang punya banyak hobi, ada yang cuma satu hobi dan ada yang tidak ada sama sekali.

g. Atribut Harus Bernilai (mandatory attribute)

Atribut harus bernilai adalah jika berisi data dan nilainya tidak boleh kosong.

Contoh : no_identitas dan nama_customer harus ada nilainya dalam tabel **customer**.

h. Atribut Nilai Null (*non-mandatory attribute*)

Atribut nilai null adalah atribut yang nilainya boleh dikosongkan. Dapat digunakan untuk menyatakan/mengisi atribut-atribut yang nilainya memang belum siap atau tidak ada. Nilai null tidak sama dengan spasi.

i. Atribut Turunan

Atribut turunan adalah atribut-atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut tabel lain yang berhubungan. Dapat

ditiadakan dari sebuah tabel, karena nilainya bergantung pada nilai yang ada di atribut lain.

c) Domain dan tipe data

Domain adalah himpunan nilai yang diijinkan pada suatu atribut. Aetiap atribut di basis data relasional didefinisikan pada suatu domain. Domain-domain dapat berbeda untuk measing-masing atribut. Dua atribut atau lebih dapat mempunyai domain yang sama. Domain tidak hanya sekedar tipe data, tetapi domain adalah tipe data yang memiliki batasan-batasan nilai tertentu.

Contoh : domain umur tidak sama dengan domain harga meskipun keduanya memiliki tipe bilangan bulat.

Tipe data	Domain
Character	Alpha Numeric (A-Z, 0-9, special character)
Number :	
▪ Integer	-32.768 s/d 32.767
▪ Real float	Pecahan
Boolean	True/false
Date	Tanggal
Memo	Text
Ole Object	Object (Image, File, Resource)

Tabel 1. Tabel tipe data inti

Menentukan tipe data

Contoh Tipe data dapat dituliskan sebagai berikut :

Atribut	Tipe	Lebar
NIM	Character	10
NAMA_MHS	Character	20

ALAMAT	Character	30
TGL_LAHIR	Date	-

Tabel 2. Struktur dan tipe data pada tabel 1

Penentuan lebar berdasarkan pada perkiraan jumlah minimal karakter yang diperlukan pada atribut tersebut. Pada contoh di atas, atribut NAMA_MHS mempunyai tipe Character dan mempunyai lebar 20, dengan alasan karena nama orang kurang dari 20 karakter. Penggunaan lebar yang kurang tepat akan berakibat pada pemborosan dan mengakibatkan file database akan membesar dengan cepat.

Pada beberapa tipe data, seperti Date dan Boolean lebar tidak perlu dicantumkan karena sudah diatur oleh DBMS.

7.2 Anomali

Anomali adalah proses pada basis data yang memberikan efek samping yang tidak diharapkan (misalnya menyebabkan ketidakkonsistenan data atau membuat suatu data menjadi hilang ketika data dihapus)

a) Anomali penyisipan

Anomali ini terjadi pada saat penambahan data ternyata ada elemen yang kosong dan elemen tsb justru menjadi key. Contoh : Tabel Kursus

NoSiswa	Kursus	Biaya
10	Bhs.Ingggris	60000
10	Bhs.Perancis	80000
10	Bhs.Jepang	70000
15	Bhs.Ingggris	60000
20	Bhs.Jepang	70000

Misalnya akan dibuka kursus baru yaitu Bhs.Jerman dengan biaya 75000 akan tetapi belum ada seorangpun yang ikut kursus ini, shg data menjadi : Tabel Kursus

NoSiswa	Kursus	Biaya
10	Bhs.Ingggris	60000
10	Bhs.Perancis	80000
10	Bhs.Jepang	70000
15	Bhs.Ingggris	60000
20	Bhs.Jepang	70000
	Bhs.Jerman	75000

b) Anomali pengubahan

Anomali ini terjadi bila ada perubahan pada sejumlah data yang mubazir, tetapi tidak seluruhnya diubah. Contoh : Tabel Pesanan

Pemasok	Kota	Barang	Jumlah
Kartika	Jakarta	Mouse	5
Citra	Bandung	Monitor	2
Yudi	Medan	CPU	2
Citra	Bandung	Printer	1

Seandainya Citra dengan kota Bandung pindah ke Bogor maka pengubahan data hanya dilakukan pada data pertama menjadi : Tabel_Pesanan

Pemasok	Kota	Barang	Jumlah
Kartika	Jakarta	Mouse	5
Citra	Bogor	Monitor	2
Yudi	Medan	CPU	2
Citra	Bandung	Printer	1

Di sini terlihat bahwa data tentang pemasok Citra tidak sama yang menyebabkan ketidakkonsistenan data.

c) Anomali penghapusan

Anomali ini terjadi apabila dalam satu baris/ tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang. Contoh pada table kursus data NoSiswa 20 akan dihapus karena sudah tidak ikut kursus lagi sehingga akibatnya data kursus bhs jepang dan biaya 70000 akan ikut terhapus.

7.3 Dependensi(ketergantungan)

Depedensi menjelaskan hubungan antar atribut, atau secara lebih khusus menjelskan nilai suatu atribut yang menentukan nilai atribut lainnya. Depedensi ini kelak menjadi acuan bagi pendekomposisian data ke dalam bentuk yang lebih efisien.

a) Dependensi fungsional

Suatu atribut Y mempunyai dependensi fungsional X jika dan hanya jika setiap nilai X berhubungan denga sebuah nilai Y.

Definisi di atas biasa di tuangkan dalam bentuk notasi sebgau berikut :

$X \rightarrow Y$ (di baca Xsecara fungsional menentukan Y).

Contoh

Pesanan_penjualan

PEMBELI	KOTA	BARANG	JUMLAH
P1	YOGYA	B1	10
P1	YOGYA	B2	5
P2	SOLO	B1	7
P2	SOLO	B2	6
P2	SOLO	B3	6
P3	KLATEN	B3	7
P3	KLATEN	B4	6

b) Dependensi sepenuhnya

Suatu atribut Y mempunyai dependensi fungsional penuh terhadap atribut X jika:

- Y mempunyai dependensi fungsional terhadap X
- Y tidak memiliki dependensi terhadap bagian dari X

Pelanggan

KODE_PLG	NAMA	KOTA	NO_FAX
P001	DINI	YOGYA	73312
P002	DINA	SOLO	75624
P003	DIAN	KLATEN	76200
P004	DINI	KLATE	76420

Pada Relasi ini :

{KODE_PLG, KOTA} → NO_FAX

KODE PLG → NO_FAX

c) Dependensi parsial

Dependensi parsial merupakan ketergantungan fungsional dimana beberapa atribut dapat di hilangkan dari A dengan ketergantungan terhadap sumber A

Contoh :

NIM	NAMA	IPK	ID_RUANG	DOSEN
A11.2078	PAIJO	3.20	301	AGUS
A112906	TUKIJO	2.02	602	JOKO
A113456	PARJO	3.40	302	RIMA
A113254	NGATIIN	3.09	301	SANTO
A11.3098	TUMINI	2.57	602	SETYO

{NIM, NAMA} \leftrightarrow ID_RUANG dimana jika nama dihilangkan maka ketergantungan tetap ada.

d) Dependensi total

Definisi dari dependensi fungsional total adalah Suatu atribut Y mempunyai dependensi Total terhadap atribut X jika;

- Y mempunyai dependensi fungsional terhadap X
- X tidak memiliki dependensi terhadap bagian dari Y

Dinyatakan dengan

Notasi: X \leftrightarrow Y

KODE_PEMASOK	NAMA_PEMASOK	KOTA
K1	KARTIKA	JAKARTA
C1	CITRA	BANDUNG
C2	CANDRA	JAKARTA

KODE_PEMASOK \leftrightarrow NAMA_PEMASOK

Dengan asumsi bahwa tidak ada nama pemasok yang sama

e) Dependensi transitif

Definisi dari dependensi Transitif adalah Suatu atribut Z mempunyai dependensi transitif terhadap atribut X jika;

- Y mempunyai dependensi fungsional terhadap X
- Z juga mempunyai dependensi fungsional terhadap Y

KULIAH	RUANG	TEMPAT	WAKTU
JARKOM	R4401	GEDUNG 4	KAMIS 09.00-10.20
MATEMATIKA	R1103	GEDUNG 1	RABU 07.00 10.00
SISTEM PAKAR	R1104	GEDUNG 1	SENIN 08.00.10.00
SIM	R4001	DEDUNG 4	SELASA 8- 8.50

Pada relasi ini

- KULIAH \rightarrow {RUANG, WAKTU}
- RUANG \rightarrow TEMPAT

Terlihat bahwa

KULIAH \rightarrow RUANG \rightarrow TEMPAT

7.4 Bentuk normal

Bentuk normalisasi

Hasil dari proses normalisasi adalah himpunan-himpunan data (table-table) dalam bentuk normal (normal form).

a) Bentuk normal pertama

Suatu relasi dikatakan bentuk normal pertama, jika dan hanya jika setiap atribut bernilai tunggal untuk setiap baris

Contoh:

Tabel_mahasiswa

NIM	Nama
12020001	Heri Susanto
12020013	Siti Zulaiha
12020015	Dini Susanti

Table_hobi

NIM	Hobi
12020001	Sepak bola
12020001	membaca komik
12020001	berenang
12020013	Memasak
12020013	menyanyi
12020015	Menjahit
12020015	membuat roti

b) Bentuk normal kedua

Suatu relasi dikatakan dalam bentuk normal ke 2 jika dan hanya jika

- Berada pada bentuk normal pertama
- Semua atribut bukan kunci memiliki dependensi sepenuhnya terhadap kunci primer

Contoh :

NIP	NAMA	JABATAN	KEAHLIAN	LAMA
107	ILHAM	ANALIS	COBOL	6
107	ILHAM	ANALIS	ORACLE	1
109	DIFA	ANALIS	CCOBOL	2
109	DIFA	ANALIS	DBASE III	2
112	RIANA	PROG	COBOL	1
112	RIANA	PROG	DBASE III	1
112	RIANA	PROG	SYBASE	1

RELASI: PEGAWAI

NIP	NAMA	JABATAN
107	ILHAM	ANALIS
109	DIFA	ANALIS
112	RIANA	PROG

RELASI : KEAHLIAN

NIP	KEAHLIAN	LAMA
107		6
107		1
109		2
109		2
112		1
112		1
112		1

c) Bentuk normal ketiga

- Berada pada bentuk normal kedua
- Setiap atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci primery

Bentuk normal kesatu:

Table pesanan

NO_PESANAN	NO_URUT	TGL_PSN	KODE_BRG	NAMA_BARANG	HARGA	TOTAL
50001	01	12-05-97	P1	PENSIL	2000	45.000
50001	02	12-05-97	P2	BUKU	3000	45.000
50001	03	12-05-97	P3	PENGGARIS	2500	45.000
50001	04	12-05-97	P4	PENGHAPUS	1000	45.000
50002	01	12-05-97	P3	PENGGARIS	2500	32.500
50002	02	12-05-97	P5	PULPEN	4000	32.500
50002	03	12-05-97	P6	SPIDOL	1500	32.500
50002	01	12-05-97	P1	PENSIL	2000	32.500
50003	02	12-05-97	P2	BUKU	3000	12.000

Bentuk normal ke 2;

Table_barang

Kode_barang	Nama_barang	Harga_satuan
p1	pensil	2000
P2	Buku	3000
P3	Panggaris	2500
P4	Penghapus	1000
P5	Pulpen	4000
P6	Spidol	1500

Tabel_pesanan

No_pesanan	Tgl_pesana	Total	No_urut	Kode_brg
50001	12-05-97	45000	01	P1
	12-05-97	45000	02	P2
	12-05-97	45000	03	P3
	12-05-97	45000	04	P4
50002	12-05-97	32500	01	P5

Bentuk normal ke 3:

Tabel_pesanan

No_pesanan*	TGL_pesanan	Total

Table_barang

Kode_item *	Nama_item	harga

Table_transaksi

No_pesanan**	No_urut	Kode_item**

d) Bentuk normal boyce-codd

Suatu relasi disebut memenuhi bentuk normal boyce-codd jika dan hanya jika suatu menentukan(determinan) adalah kunci kandidat (atribut yang bersifat unik).

Relasi KURSUS

SISWA	KURSUS	TUTOR
ANWAR	B PRANCIS	PIERRI
ANWAR	B INGGRIS	PIERRI
BUDI	B PRANCIS	PIERRI
CECEP	B INGGRIS	PIERRI

Cara mengkonfirmasi relasi yang telah menentukan bentuknormal ke 3 ke BCNF adalah:

- Carilah semua penentu
- Bila terdapat penentu bukan berupa kunci kandidat, maka

- Pisahkan relasi tersebut
- Buat penentu tersebut sebagai kunci primer

Siswa	tutor

Tutor	kursus

e) Bentuk normal keempat

Suatu relasi memenuhi bentuk normal ke empat

- Telah berada pada BCNF
- Tidak mengunkan dua atribut lebihyang bernlai banyak

MATAKULIAH	DOSEN	ISI
PENGETAHUAN KOMPUTER	AMIR FITRIA	DASAR KOMPUTER PENGENALAN PENGOLAHAN KATA PENGENALAN LEMBAR KERJA
MATEATIKA 1	FITRIA	DIFERENSIAL INTEGRAL

RELASI yang telah di normalkan berdasarkan table di atas.

MATA KULIAH	DOSEN	ISI
PENGETAHUAN KOMPUTER	AMIR	DASAR KOMPUTER
PENGETAHUAN KOMPUTER	AMIR	PENGANTAR PENGOLAHAN KATA
PENGETAHUAN KOMPUTER	AMIR	PENGANTAR LEMBAR KERJA
PENGETAHUAN KOMPUTER	FITRIA	DASAR KOMPUTER

Oleh karena itu relasi di atas dapat didekomposisikan menjadi tiga relasi;

- DEAL_DIST(DEALER,DISTRIB)
- DIST_KEND (DISTRIB,KENDARAAN)
- DEAL_KEND (DEALER, KENDARAAN)

8. Praktikum 1: Normalisasi Data

8.1 Analisis sistem informasi

Pada pertemuan ke 8 ini dosen akan memberikan soal tentang normalisasi,

Soal normalisasi nya adalah

Buat lah design conpetual schema dari dokumen dasar berikut:

BORMA DAGO TOSERBA	
Jl.Ir.H. Juanda 348, BANDUNG	
TLP 2504601	
<hr/>	
Tgl	: 10-10-2010
Jam	: 10.26.12
Kasir	: Insan (32A)
<hr/>	
Beras merah 1 KG/PRIMA	19.600
2x9.800	
WETKINS WIPES 10S/PURPLE	2.800
1X2.800	
WETKING WIPES 10S/YE	2.800
1X2.800	
INDOMIE AYAM BAWANG 69 GR	4.800
4X1.200	
UPET THE 35 GR	2.200
1X2.200	
SW-31 FRESH SEALWARE	8.800
1X8.500	
UPET THE 35 GR	2.200
1X2.200	
<hr/>	
TOTAL	42.900
BAYAR	100000
<hr/>	
KEMBALI	57.100
NO BON	: P1010.1848
<hr/>	

TERIMA KASIH ATAS KUNJUNGAN ANDA

8.2 Normalisasi data

Buat bentuk normalisasi

- Bentuk unnormalisasi
- Normalisasi ke 1
- Normalisasi ke 2
- Bentuk normal boyce-codd
- Bentuk normal keempat

9. Pemodelan Data

Merupakan sekumpulan konsep untuk menggambarkan data, hubungan antar data, semantik (makna) data dan batasan data.

9.1 Model Entity-Relationship (E-R)

Menurut Fathansyah (2012:72) model data didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data. Ada sejumlah cara dalam merepresentasikan model data dalam perancangan basis data, yang salah satunya adalah model E-R. Dalam perancangan konseptual basis data diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antardata. Hubungan tersebut dapat dinyatakan dalam bentuk model E-R. Menurut Kadir, A. (2009:30) model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antara entitas. Model ini dinyatakan dalam bentuk diagram. Itulah sebabnya model E-R disebut sebagai diagram E-R.

Terdapat dua komponen utama yang membentuk model E-R yaitu Entitas (Entity) dan Relasi (Relationship). Kedua komponen ini dideskripsikan lebih jauh melalui sejumlah Atribut.

a) Entitas

Entitas dapat berupa sesuatu yang nyata atau abstrak (berupa suatu konsep). Secara lebih rinci, Hoffer, dkk (2005) menjelaskan bahwa entitas dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep. Berikut beberapa contoh entitas:

1. Contoh entitas dari orang: PEGAWAI, MAHASISWA, PASIEN
2. Contoh entitas dari tempat : TOKO, GUDANG, PROVINSI

3. Contoh entitas dari objek : MESIN, GEDUNG, MOBIL
4. Contoh entitas dari kejadian : PENJUALAN, REGISTRASI
5. Contoh entitas dari konsep : REKENING, KURSU

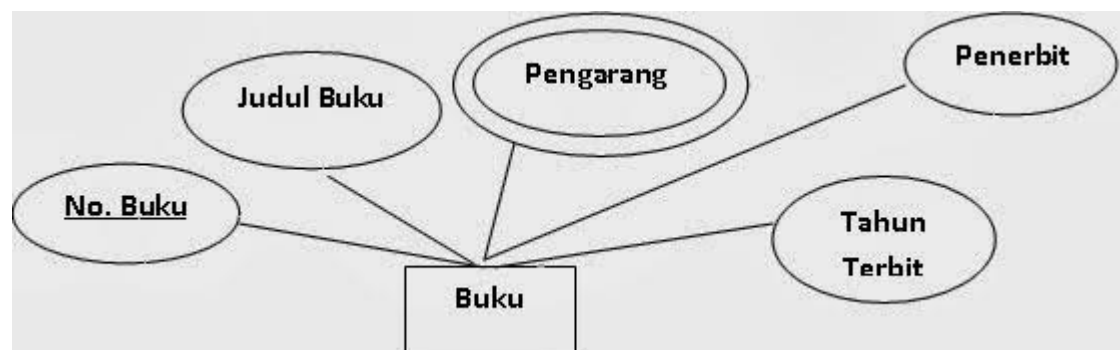
Pada diagram E-R, sebuah entitas digambarkan dengan simbol berupa persegi panjang dengan nama entitas di dalamnya yang ditulis dengan huruf kapital. Contohnya dapat dilihat pada gambar 1 di bawah ini



Gambar 1. Penggambaran Entitas Dalam Model/Diagram E-R Pada bahasan varian entitas akan diperlihatkan beberapa variasi penggambaran entitas, bergantung pada jenisnya.

b) Atribut

Setiap entitas memiliki sejumlah atribut. Atribut merupakan properti atau karakteristik yang mendeskripsikan sebuah entitas. Sebagai contoh entitas mahasiswa memiliki beberapa atribut diantaranya: Nim, Nama, Tanggal_Lahir, Jenis_Kelamin. Pada diagram E-R, sebuah atribut digambarkan dengan simbol lingkaran lonjong (elips) dengan nama atribut di dalamnya yang ditulis menggunakan huruf kapital untuk setiap awal kata dan dipisahkan oleh karakter garis bawah (_) apabila lebih dari satu kata.



Jenis-jenis atribut :

- Atribut Key

Atribut Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (Row/Record) dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan key tidak boleh ada baris data dengan nilai yang sama

Contoh : Nomor pokok mahasiswa (NPM), NIM dan nomor pokok lainnya

- Atribut simple

atribut yang bernilai atomic, tidak dapat dipecah/ dipilah lagi

Contoh : Alamat, penerbit, tahun terbit, judul buku.

- Atribut Multivalued

nilai dari suatu attribute yang mempunyai lebih dari satu (multivalued) nilai dari attribute yang bersangkutan

Contoh : dari sebuah buku, yaitu terdapat beberapa pengarang.

- Atribut Composite

Atribut composite adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisa dipecah lagi atau mempunyai sub attribute.

Contoh : dari entitas nama yaitu nama depan, nama tengah, dan nama belakang

1. Atribut Derivatif

Atribut yang tidak harus disimpan dalam database Ex. Total. atau atribut yang dihasilkan dari atribut lain atau dari suatu relationship.

Atribut ini dilambangkan dengan bentuk oval yang bergaris putus-putus

c) Relasi

Relasi pada model/diagram E-R memiliki definisi yang tidak sama pada istilah yang ada di dalam basis data relasional. Dimana relasi dalam basis data relasional memiliki arti berupa tabel yang terdiri atas baris dan kolom. Di dalam model E-R relasi merupakan hubungan atau keterkaitan antara beberapa entitas. Sebagai contoh

bahwa entitas MAHASISWA memiliki relasi dengan entitas DOSEN dalam hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing.

Simbol : diamond/belah ketupat

→ Contoh :



d) Kamus Data

Tujuan utama dari pembuatan model/diagram E-R adalah untuk menunjukkan objek-objek (entitas) apa saja yang ingin dilibatkan dalam sebuah basis data dan bagaimana hubungan yang terjadi diantara entitas tersebut. Pada sebuah diagram E-R kadang kala penggambaran atribut-atribut pada sebuah entitas atau relasi dapat mengganggu tujuan yang ingin dicapai. Maka dari itu kita dapat memisahkan pendeklarasian atribut-atribut ini dari diagram E-R dan menyatakannya dalam sebuah kamus data. Kamus data berisi daftar atribut yang diapit kurung kurawal (“{“ dan “}”). Atribut yang berfungsi sebagai kunci (key) dengan menggarisbawahi atribut tersebut. Berikut contoh diagram E-R dengan menggunakan Kamus Data:



Kamus Data:

- Mahasiswa = {nim, nama_mhs, alamat_mhs dan tgl_lahir}
- Kuliah = {kode_kul, nama_kul, sks dan semester}
- Dosen = {nama_dos, alamat_dos}
- Mempelajari = {nim, kode_kul, indeks_nilai}
- Mengajar = {kode_kul, nama_dos, waktu, tempat}

9.2 Varian entitas

Suatu entitas dapat diklasifikasikan menjadi dua jenis yaitu diantaranya:

1. Entitas Kuat (Himpunan Entitas Kuat/Tipe Entitas Kuat/Strong Entity Set)

Entitas kuat merupakan entitas yang keberadaannya tidak bergantung pada entitas yang lain. Contohnya entitas MAHASISWA, DOSEN, MATAKULIAH tergolong sebagai entitas kuat, dimana kemunculan instan-instan di dalamnya tidak bergantung pada keberadaan instan-instan di entitas yang lain. Entitas tersebut juga bukan merupakan bagian (sub) dari entitas yang lain. Entitas kuat selalu memiliki atribut pengenal atau kunci (key), entah itu tersusun atas sebuah atribut atau beberapa atribut.

2. Entitas Lemah (Himpunan Entitas Lemah/Tipe Entitas Lemah/Weak Entity Set)

Sedangkan entitas lemah merupakan entitas yang keberadaannya bergantung pada entitas lain. Entitas lemah tidak memiliki atribut pengenal atau kunci (key), yang benar-benar dapat menjamin keunikan entitas di dalamnya. Sebagai contoh, entitas ORANGTUA dan entitas HOBBY berisi data orang tua yang dimiliki mahasiswa, serta hobby yang disenangi mahasiswa. Keberadaan kedua entitas tersebut bergantung pada entitas MAHASISWA

9.3 Varian atribut

Atribut yang digunakan dalam model/diagram E-R dapat dikategorikan sebagai berikut:

1. Atribut kunci/pengenal dan atribut deskriptif
2. Atribut sederhana dan atribut komposit.
3. Atribut bernilai tunggal dan atribut bernilai banyak
4. Atribut tersimpan dan atribut turunan

9.4 Varian relasi

Relasi yang terjadi di antara dua himpunan entitas yang berbeda disebut sebagai Relasi Biner (binary relation), yang merupakan relasi yang paling umum digunakan. Namun, demikian ada kalanya kita juga menggunakan relasi yang hanya melibatkan sebuah himpunan entitas atau lebih dari dua himpunan entitas.

9.5 Spesialisasi dan generalisasi

Jika kita memulai dari sebuah himpunan entitas lalu demikian melakukan pengelompokan yang melahirkan himpunan entitas baru (proses top down). Maka kita sedang melakukan Spesialisasi. Bisa juga yang terjadi adalah sebaliknya. Kita mengetahui bahwa entitas-entitas dalam himpunan entitas mahasiswa sebenarnya dapat dibagi dalam dua kelompok, yaitu mahasiswa D3 dan mahasiswa S1 tetapi pengelompokan ini tidak dipertegas dengan adanya perbedaan atribut. Karena tidak tegasnya perbedaan atribut dari dua kelompok, maka kelompok-kelompok entitas tersebut malah disatukan dalam sebuah himpunan entitas dengan atribut-atribut yang sama. Jadi pendekatannya bersikap Bottom Up, mula-mula terpisah tetapi kemudian menjadi satu. Proses yang demikian disebut generalisasi. Dengan demikian spesialisasi dan generalisasi merupakan dua proses yang berlawanan. Yang ditekankan dalam spesialisasi adalah perbedaan antar dua kelompok entitas, sedangkan dalam generalisasi diwujudkan dalam notasi relasi yang khusus disebut relasi 'ISA' (yang berasal dari 'Is A').

9.6 Agregasi

Agregasi dalam SQL merupakan proses untuk mendapatkan nilai dari sekumpulan data yang telah dikelompokkan. Pengelompokan data didasarkan pada kolom atau kombinasi kolom yang dipilih.

Beberapa fungsi untuk agregasi adalah:

1. MAX : mencari data terbesar dari sekelompok data
2. MIN : mencari data terkecil dari sekelompok data
3. COUNT : mencari cacah data (data NULL tidak akan dimasukkan dalam perhitungan, kecuali disebutkan secara khusus)
4. SUM : mencari jumlah dari sekumpulan data numeris
5. AVG : mencari rerata dari sekumpulan data numeris

9.7 Asosiatif

yaitu gabungan antara hubungan dan entitas. Hubungan dengan Atribut, Mendeskripsikan hal – hal yang berkenaan dengan asosiasi antar entitas dalam hubungan tersebut.



9.8 Tahapan pembuatan diagram Entity-Relationship (E-R)

Berikut ini adalah contoh tahapan dalam membuat ERD pada Sistem Informasi Akademik:

Tahap 1: Penentuan Entities



Tahap 2 : Penentuan Atribut

Mahasiswa:

- nim: nomor induk mahasiswa (integer) PK
- nama_mhs: nama lengkap mahasiswa (string)
- alamat_mhs: alamat lengkap mahasiswa (string)

Dosen:

- nip: nomor induk pegawai (integer) PK

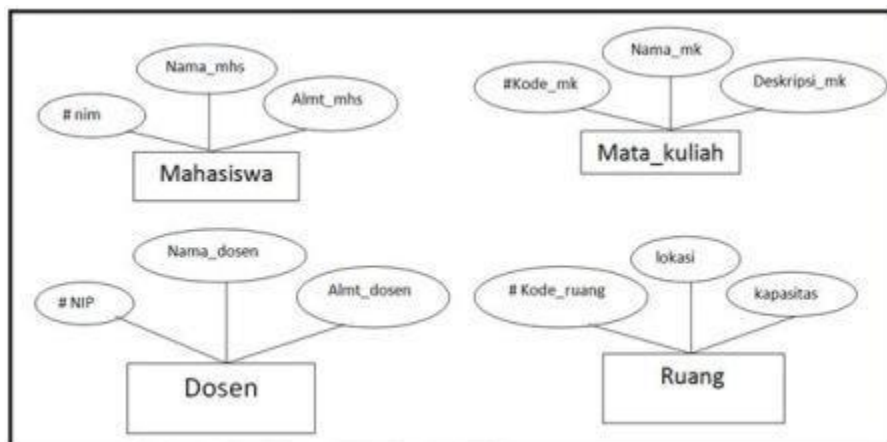
- nama_dosen: nama lengkap dosen (string)
- alamat_dosen: alamat lengkap dosen (string)

Mata_kuliah:

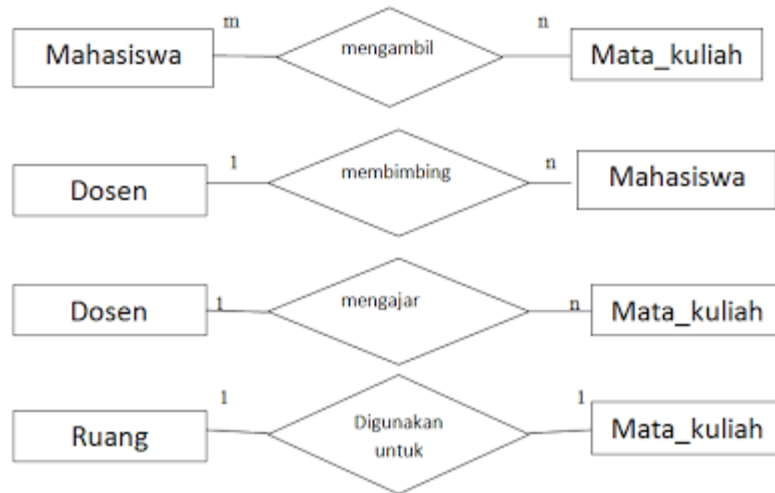
- kode_mk: kode untuk mata kuliah (integer) PK
- nama_mk: nama lengkap mata kuliah (string)
- deskripsi_mk: deskripsi singkat mengenai mata kuliah (string)

Ruang:

- kode_ruang: kode untuk ruang kelas (string) PK
- lokasi_ruang: deskripsi singkat mengenai lokasi ruang kelas (string)
- kapasitas_ruang: banyaknya mahasiswa yang dapat ditampung (integer)



Tahap 3: Penentuan Kardinalitas Relasi



Hubungan :

a. ruang digunakan untuk mata_kuliah:

- Tabel utama: ruang
- Tabel kedua: mata_kuliah
- Relationship: One-to-one (1:1)
- Attribute penghubung: kode_ruang (FK kode_ruang di mata_kuliah)

b. dosen mengajar mata_kuliah:

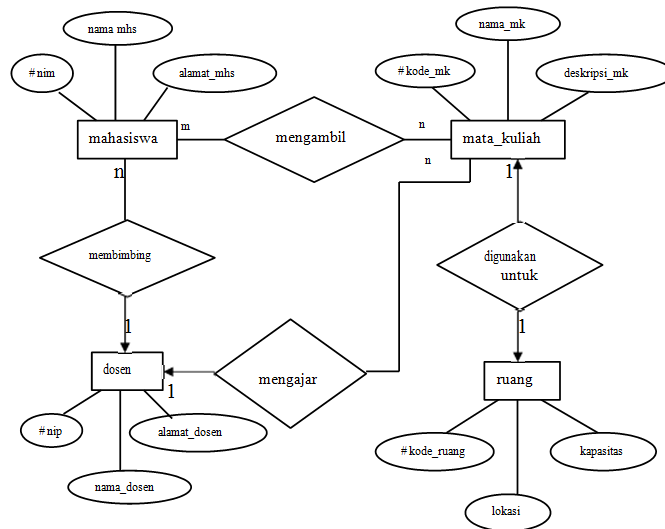
- Tabel utama: dosen
- Tabel kedua: mata_kuliah
- Relationship: One-to-many (1:n)
- Attribute penghubung: nip (FK nip di mata_kuliah)

c. dosen membimbing mahasiswa:

- Tabel utama: dosen
- Tabel kedua: mahasiswa
- Relationship: One-to-many (1:n)
- Attribute penghubung: nip (FK nip di mahasiswa)

- d. mahasiswa mengambil mata_kuliah:
- Tabel utama: mahasiswa, mata_kuliah
 - Tabel kedua: mhs_ambil_mk
 - Relationship: Many-to-many (m:n)
 - Attribute penghubung: nim, kode_mk (FK nim, kode_mk di mhs_ambil_mk)

Tahap 4 : Pembuatan ERD



10. Praktikum 2: Pemodelan Data

10.1 Normalisasi data hasil praktikum 1

Pada pertemuan ke 10 ini dosen akan memberikan soal tentang normalisasi,

Soal normalisasi nya adalah

Buat lah design conpetual schema dari dokumen dasar berikut:

10.2 Pembangunan diagram Entity-Relationship (E-R)

BORMA DAGO TOSERBA
Jl.Ir.H. Juanda 348, BANDUNG
TLP 2504601

Tgl : 10-10-2010

Jam : 10.26.12

Kasir : Insan (32A)

Beras merah 1 KG/PRIMA	19.600
------------------------	--------

2x9.800

WETKINS WIPES 10S/PURPLE	2.800
--------------------------	-------

1X2.800

WETKING WIPES 10S/YE	2.800
----------------------	-------

1X2.800

INDOMIE AYAM BAWANG 69 GR	4.800
---------------------------	-------

4X1.200

UPET THE 35 GR	2.200
----------------	-------

1X2.200

SW-31 FRESH SEALWARE	8.800
----------------------	-------

1X8.500

UPET THE 35 GR	2.200
----------------	-------

1X2.200

TOTAL	42.900
-------	--------

BAYAR	100000
-------	--------

KEMBALI	57.100
---------	--------

NO BON : P1010.1848

TERIMA KASIH ATAS KUNJUNGAN ANDA

11. Ujian Tengah Semester

12. Penerapan Basis Data

12.1 Relasi table

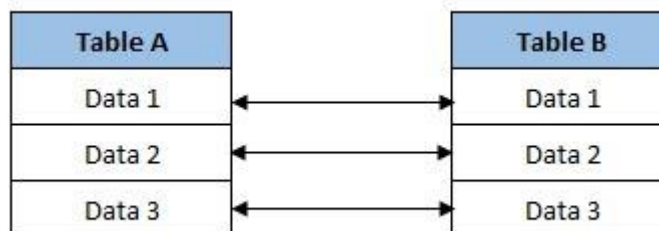
Relasi tabel adalah hubungan sebuah tabel dengan tabel lainnya. Sehingga tabel tidak lagi berdiri sendiri, melainkan dapat dihubungkan antara satu dengan yang lainnya dan menjadi satu kesatuan.

Perlu diketahui bahwa sebuah tabel hanya boleh memiliki satu buah primary key (kunci utama). Namun, sebuah tabel boleh memiliki lebih dari satu buah foreign key (kunci asing). Oleh karena itu, pilihlah satu buah kolom pada tabel yang akan dijadikan primary key yang dapat mewakili kolom lainnya dan nilainya pun unik, misalnya kolom nik (nomor induk karyawan), nim (nomor induk mahasiswa) dan lainnya.

Berikut ini adalah 3 jenis relasi tabel database diantaranya, yaitu:

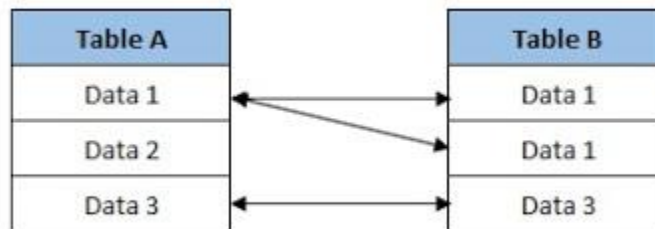
1. One to One (1 to 1)

Relasi One to One (1 to 1) terjadi apabila sebuah data yang ada pada tabel A juga ada pada tabel B, yang mana data tersebut hanya diperbolehkan satu buah saja pada masing-masing tabel, tidak boleh terdapat beberapa data yang sama pada tabel yang sama.



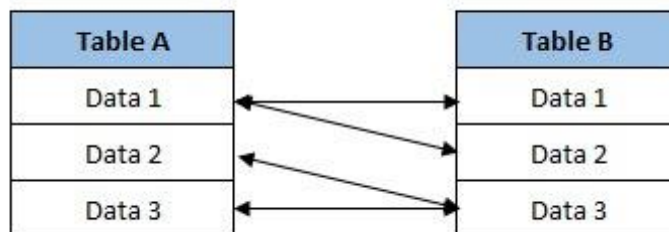
2. One to Many (1 to n)

Relasi One to Many (1 to n) terjadi apabila sebuah data yang ada pada tabel A memiliki beberapa data yang sama pada tabel B. Pada relasi ini hanya diperbolehkan sebuah data pada tabel A dan tabel B boleh memiliki beberapa data yang sama pada tabel A.



3. Many to Many (n to m)

Relasi Many to Many (n to m) terjadi apabila sebuah data yang ada pada tabel A memiliki beberapa data yang sama pada tabel B dan sebuah data pada tabel B juga memiliki beberapa data yang sama pada tabel A. Pada relasi ini tabel A dan tabel B diperbolehkan memiliki beberapa data yang sama dengan kedua tabel tersebut.



13. Denormalisasi Basis Data

13.1 Redundansi dan bentuk-bentuk denormalisasi

Perbedaan normalisasi dan denormalisasi adalah terletak pada redundansi data dan kompleksitas query. Pada redundansi data normalisasi lebih strik atau harus dihilangkan sebisa mungkin sehingga mengakibatkan apabila kita akan mengakses data dalam suatu database membutuhkan query yang kompleks. Berbeda dengan denormalisasi, denormalisasi disini tidak terlalu memikirkan tentang data yang redundan sehingga dalam mengakses data lebih cepat. Dan juga pada normalisasi integritas data akan meningkat sedangkan pada denormalisasi, integritas data akan menurun.

Bentuk-bentuk Denormalisasi

1. Atribut yang terderivasi (atribut turunan)

Atribut turunan adalah atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan. Atribut yang demikian sebenarnya dapat ditiadakan dari sebuah tabel, karena nilai-nilainya bergantung pada nilai yang ada pada atribut lainnya

2. Atribut yang berlebihan

3. Atribut Gabungan

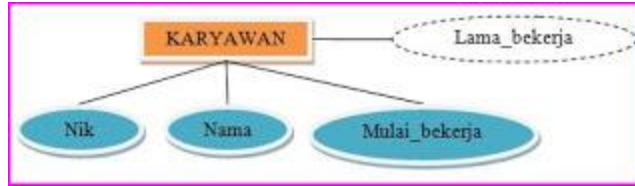
13.2 Atribut turunan

Yaitu atribut yang dapat diperoleh dari nilai suatu atribut yang tersimpan dalam basis data.

Contoh :

Atribut Usia, dapat diperoleh dari atribut tanggal lahir yang tersimpan dalam basis data.

Atribut Lama_bekerja, dapat diperoleh dari atribut Tanggal_mulai_bekerja yang tersimpan dalam basis data.



13.3 Atribut yang berlebihan

Atribut yang berlebihan atribut yang menyatakan lebih dari satu fakta. Atribut berlebihan terbagi atas beberapa bagian yaitu :

- **Atribut Terkodekan**

Atribut terkodekan adalah atribut yang memiliki kode tambahan yang menunjukkan beberapa kondisi lainnya.

Contoh :

id_mk di tabelkuliah yang didalamnya sudah terkandung data program studi. Data program studi ini sebenarnya tidak diperlukan lagi karena sudah ada atribut prog_studi di tabel kuliah.

Tapi akan menjadi aneh jika kode matakuliah tidak mengikuti format aturan penulisan yang ada. Untuk itu kita bisa lakukan denormalisasi dengan tetap menuliskan kode matakuliah seperti di atas.

- **Atribut Gabungan**

Atribut gabungan adalah atribut yang terdiri dari atribut yang lain.

Contoh :

Atribut *nim* di tabel mahasiswa merupakan gabungan dari tahun masuk/angkatan dengan program studi dan no urut mahasiswa. Dengan demikian atribut ini sebenarnya tidak atomik (bagian terkecil) karena masih bisa dibagi lagi.

Misalnya : NIM : 09.51.0062 (Ket: angkatan.jurusan.no urut).

- **Atribut Tumpang Tindih**

Atribut tumpang tindih adalah atribut dengan nilai yang tidak sepenuhnya eksklusif (bersifat khusus).

Contoh :

Atribut semester di tabel kuliah berisikan:

- ‘1’ : matakuliah ganjil
- ‘2’ : matakuliah genap
- ‘3’ : matakuliah ganjil & genap

Nilai ‘3’ mencakup semester genap dan ganjil sekaligus(jadi tidak eksklusif).

Jadi sebaiknya matakuliah dimasukkan dua kali, jika itu di laksanakan di semester ganjil dan genap.

13.4 Tabel rekapitulasi

Laporan hasil rekapitulasi akan selalu merupakan hasil pengolahan dari semua tabel yang ada. Pengolahan tersebut melibatkan banyak tabel sehingga akan membutuhkan waktu yang lama.

Jika hal tersebut sering diakses dan diperlukan, maka perlu dibuat tabel khusus untuk menyimpan data hasil rekapitulasi tersebut.

Pada contoh kasus sebelumnya, akan dibutuhkan waktu yang lama jika harus menghitung jumlah matakuliah, jumlah sks dan ipk mahasiswa yang pengolahannya berasal dari beberapa tabel.

Untuk itu bisa dibuat tabel khusus, misal : rekap_mahasiswa yang berisi data tentang jumlah matakuliah, jumlah sks, ipk.

Hal ini tentu saja akan menimbulkan redundansi, tapi dengan mempertimbangkan performansi, Denormalisasi pada kasus ini perlu dilakukan.

14. Bahasa Query

14.1 Data Definition Language (DDL)

merupakan kumpulan perintah SQL yang digunakan untuk membuat, mengubah dan menghapus struktur dan definisimetadata dari objek-objek database.

Objek-objek database pada yang dimaksud - pada MySQL - adalah sebagai berikut :

- Database
- Table
- View
- Index
- Procedure (Stored Procedure)
- Function
- Trigger

Contoh :

```
-- Perintah berikut akan membuat table pada database kita dengan nama  
"ms_karyawan"
```

```
CREATE TABLE ms_karyawan (  
  kode_cabang varchar(10) default NULL,  
  kode_karyawan varchar(10) NOT NULL,  
  nama_depan varchar(8) default NULL,  
  nama_belakang varchar(9) default NULL,  
  jenis_kelamin varchar(1) default NULL,  
  PRIMARY KEY (kode_karyawan)  
)
```

Berikut adalah contoh perintah DDL yang digunakan pada MySQL. Klik pada link untuk melihat rincian penggunaan dari perintah tersebut:

- **Pembuatan (CREATE)**
 - CREATE DATABASE
 - CREATE FUNCTION
 - CREATE INDEX
 - CREATE PROCEDURE
 - CREATE TABLE
 - CREATE TRIGGER
 - CREATE VIEW

14.2 Data Manipulation Language (DML)

merupakan kumpulan perintah SQL yang digunakan untuk **proses pengolahan isi data** di dalam table seperti memasukkan, merubah dan menghapus **isi data** - dan tidak terkait dengan perubahan struktur dan definisi tipe data dari objek database.

Contoh :

```
INSERT INTO ms_karyawan(
    kode_cabang,
    kode_karyawan,
    nama_depan,
    nama_belakang,
    jenis_kelamin)
VALUES ('cab-001','kar-001','Feris','Thia','Laki-laki');
```

Beberapa daftar Perintah DML MySQL 5.0

- CALL
- DELETE
- DO
- HANDLER
- INSERT
- LOAD DATA INFILE
- REPLACE
- SELECT
- TRUNCATE
- UPDATE

15. Pengamanan Basis Data

15.1 Perintah SQL untuk pengamanan basis data

Perintah utama pemberian otorisasi ini adalah: Grant dengan syntax :Grant<daftar otoritas> on < nama objek basis data> To <daftar user>

- <daftar otoritas>: klausa otoritas

- < nama objek basis data>: dapat berupa tabel/view

- <daftar user>: User pemakai

Contoh: Grant insert on Mahasiswa to ali, ani

15.2 Penyandian (Enkripsi)

Teknik enkripsi yang baik memiliki kriteria: Pemakai yang memiliki otoritas dapat dengan mudah melakukan penyandian dan pemecahannya. Skema enkripsi yang digunakan tidak tergantung kerahasiaan algoritma /program enkripsi dan dekripsi tetapi tergantung pada parameter algoritma sebagai kunci enkripsinya. Sangat sukar untuk ditebak kunci enkripsi yang digunakan.

15.3 Pemeliharaan integritas basis data

Sebagai sarana untuk meyakinkan bahwa nilai-nilai data dalam sistem basis data selalu benar, konsisten, selalu tersedia. Dapat dilakukan dengan cara :

- Pastikan bahwa nilai-nilai data adalah benar sejak dimasukkan pertama kali
- Membuat program untuk mengecek keabsahan data pada saat dimasukkan ke computer
- Penolakan / pembatalan aksi (cancelation)
- Pengisian nilai kosong pada field tertentu (**nullify**)
- Penjalaran perubahan (cascade)

16. Persentasi Tugas Kelompok

16.1 Pengumpulan laporan perancangan basis data

Mahasiswa di bagi menjadi beberapa kelompok untuk mengerjakan soal yang telah di berikan

PT. PUJASARI Jl. Letjen. MT. Haryono 23 Telp. (0274)378453 YOGYAKARTA		No. SOP	:	688	
SURAT ORDER PENGIRIMAN (SOP)		Tanggal	:	15 Juni 2007	
		No. Order	:	4 560 654 079	
		Tanggal Order	:	10 Juni 2007	
		Syarat Penyerahan	:	Loco Gudang Pembeli	
		Syarat Pembayaran	:	2/10 n/30	
		Dijual Kepada	:	PT. Putra Surya	
Alamat	:	Jl. Raya Demak No.59 Telp. (0291) 721165 Jepara			
Dikirim Kepada	:	PT. Putra Surya			
Alamat	:	Jl. Raya Demak No.59 Telp. (0291) 721165 Jepara			
Tanggal Pengiriman	:	15 Juni 2007			
Rute Pengiriman	:	Jogya-Semarang-Jepara			
No. Urut	Keterangan	Satuan	Kuantitas	Harga Satuan	Jumlah Harga
1	Kertas HVS A 4 70 gr	Rim	240	28.000	6.720.000
2	Kertas HVS A5 70 gr	Rim	300	24.000	7.200.000
3	Kertas HVS F4 70 gr	Rim	150	34.000	5.100.000
				Jumlah	19.020.000
Bagian Pengiriman		Bagian Kredit		Bagian Order Perjualan	
Sukardi		Melansari		Waryanto	

16.2 Persentasi hasil perancangan basis data

Mahasiswa harus mempresentasikan hasil diskusi merka

16.3 Tanya jawab

18. Persentasi Tugas Kelompok

18.1 Pengumpulan laporan perancangan basis data

Mahasiswa di bagi menjadi beberapa kelompok untuk mengerjakan soal yang telah di berikan

PT. PUJASARI Jl. Letjen. MT. Haryono 23 Telp. (0274)378453 YOGYAKARTA		No. SOP : 688			
SURAT ORDER PENGIRIMAN (SOP)		Tanggal : 15 Juni 2007			
		No. Order : 4 560 654 079			
		Tanggal Order : 10 Juni 2007			
		Syarat Penyerahan : Loco Gudang Pembali			
		Syarat Pembayaran : 2/10 n/30			
Dijual Kepada : PT. Putra Surya					
Alamat : Jl. Raya Demak No.59 Telp. (0291) 721165 Jepara					
Dikirim Kepada : PT. Putra Surya					
Alamat : Jl. Raya Demak No.59 Telp. (0291) 721165 Jepara					
Tanggal Pengiriman : 15 Juni 2007					
Rute Pengiriman : Jogja-Semarang-Jepara					
No. Urut	Keterangan	Satuan	Kuantitas	Harga Satuan	Jumlah Harga
1	Kertas HVS A 4 70 gr	Rim	240	28.000	6.720.000
2	Kertas HVS A5 70 gr	Rim	300	24.000	7.200.000
3	Kertas HVS F4 70 gr	Rim	150	34.000	5.100.000
Jumlah					19.020.000
Bagian Pengiriman Sukardi		Bagian Kredit Melansari		Bagian Order Penjualan Waryanto	

18.2 Persentasi hasil perancangan basis data

Mahasiswa harus mempresentasikan hasil diskusi merka

18.3 Tanya jawab

Daftar pustaka

<http://fairuzelsaid.com/jenis-jenis-atribut-kunci-key-field-dalam-basis-data/>

<https://fairuzelsaid.wordpress.com/2010/01/24/sistem-basis-data-normalisasi/>

<https://dosenit.com/kuliah-it/database/komponen-database>

<http://slideplayer.info/slide/3991597/>