

BAB I**KONSEP DASAR DAN DEFINISI BASIS DATA****Tujuan :**

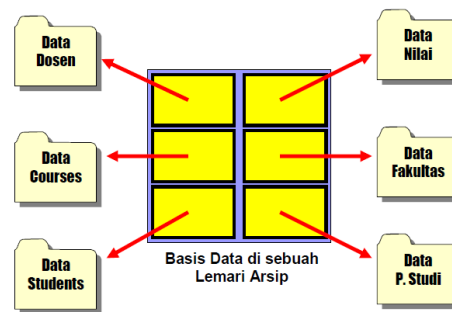
1. Mengetahui konsep dasar dan definisi basis data
2. Mengetahui komponen basis data
3. Mengetahui keuntungan dan kerugian basis data
4. Mengetahui penggunaan basis data

1.1 Konsep Dasar dan Definisi

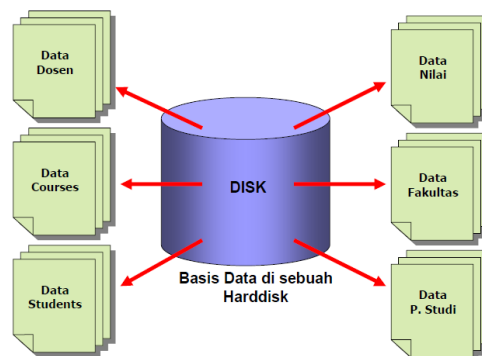
Sebenarnya basis data tidak selalu berhubungan dengan komputer karena pencatatan secara manual ke dalam suatu buku besar pun bisa dianggap sebagai suatu basis data, akan tetapi saat ini bila berbicara mengenai teknologi basis data maka akan selalu dihubungkan dengan teknologi komputer karena kedua hal tersebut berjalan beriringan.

Banyak definisi yang digunakan untuk menjelaskan istilah basis data yaitu :

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali secara cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- Kumpulan file/tabel/arsip/dokumen yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.
- Basis Data (*Database*), dapat dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki lemari arsip dan berwenang/bertugas untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal-hal seperti : memberi sampul/map pada kumpulan/bundel arsip yang akan disimpan, menentukan kelompok/jenis arsip, memberi nomor dengan pola tertentu yang nilainya unik pada setiap sampul/map.



Gambar 1. basis data di sebuah lemari arsip



Gambar 2. basis data di sebuah harddisk

1.2 Komponen Basis Data

1. Data (Data Operasional, Data Masukan, Data Keluaran)
2. Hardware (Perangkat Keras)
3. Software (Perangkat Lunak)
4. User / Pemakai

Keterangan :

1. Data

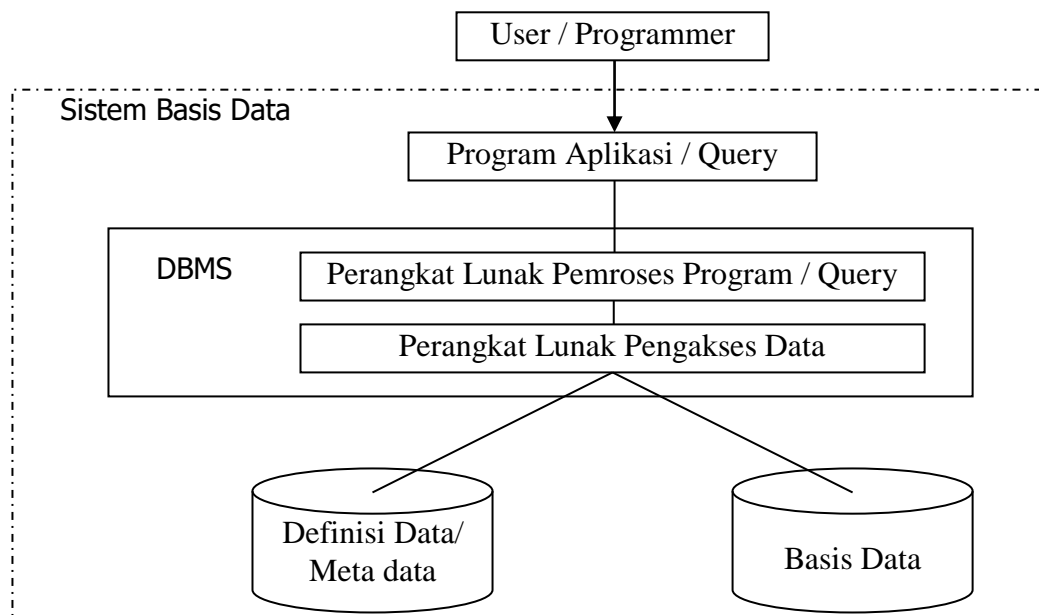
- **Data Operasional** : yaitu data yang disimpan dalam basis data baik itu berupa data master maupun data transaksi.
- **Data Masukan (Input data)** : yaitu data dari luar sistem yang dimasukkan melalui peralatan input seperti keyboard yang dapat mengubah data operasional.
- **Data Keluaran (Output data)** : yaitu data berupa laporan melalui peralatan output seperti screen, printer dll, sebagai hasil proses dari dalam suatu sistem yang mengakses data operasional.

2. **Hardware (Perangkat Keras)** : Terdiri dari semua peralatan komputer yang berbentuk fisik yang digunakan untuk pengelolaan basis data, seperti : HardDisk, Keyboard, Monitor dll.
3. **Software (Perangkat Lunak)** : Digunakan sebagai interface (antar Muka) antara pemakai dengan data fisik basis data, misalnya : Program-program aplikasi (POS, Inventory System dll) maupun DBMS (MySQL, SQL Server, Oracle dll).
4. **User atau Pemakai** dibagi menjadi 4 bagian :
 1. Naïve User / Pemakai Awam
Yaitu pemakai yang tidak berpengalaman / tidak mahir. Berinteraksi dengan basis data menggunakan program aplikasi yang khusus dibuat untuk suatu keperluan. Misal : Teller Bank, Kasir, Web User.
 2. Sophisticated User / Pengguna Canggih
Yaitu Pemakai yang berinteraksi dengan basis data tidak melalui program melainkan menggunakan bahasa query. Misal : Analis sistem, Pelaku End User Computing.
 3. Application Programmer / Pemrogram Aplikasi
Yaitu profesional dibidang komputer yang membuat program aplikasi menggunakan tool untuk membuat program aplikasi seperti Visual Foxpro, Delphi dll.
 4. Data Base Administrator (DBA)
Yaitu orang atau tim yang bertugas untuk mengelola sistem basis data secara keseluruhan. Tugas dari DBA antara lain :
 - a. Mendefinisikan maupun modifikasi Skema (dengan pernyataan DDL)
 - b. Mendefinisikan struktur tempat penyimpanan dan metode akses
 - c. Mendefinisikan Batasan Integritas
 - d. Pemberian Kewenangan Hak Akses
 - e. Pemeliharaan (Backup, recovery, memantau kinerja)

1.3 Sistem Basis Data

Sistem Basis Data adalah gabungan antara Basis data dan perangkat lunak DBMS termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam

suatu sistem yang bertujuan untuk dapat memanipulasi data dari basis data sehingga diperoleh informasi yang diinginkan.



Gambar 3. Gambaran Lingkungan Sistem Basis Data

Terdapat perbedaan antara pendekatan pemrosesan berkas dan pendekatan pemrosesan basis data. Dalam pendekatan basis data sejumlah data disimpan dalam satu tempat dengan definisi data yang tetap sehingga dapat diakses oleh beberapa pemakai dengan berbagai program aplikasi. Definisi data disimpan dalam sistem katalog (terpisah dari data → *program-data independent*) yang berisi informasi tentang struktur, tipe dan format penyimpanan data. Informasi ini disebut meta data. Meta data digunakan oleh DBMS untuk mengakses data.

Pada pendekatan pemrosesan berkas, struktur data didefinisikan dalam program aplikasi sehingga hanya program aplikasi tersebut yang dapat mengakses datanya.

1.4 Keuntungan dan Kerugian Basis Data

Keuntungan pendekatan basis data

Berikut ini beberapa keuntungan menggunakan pendekatan basis data secara elektronis :

1. Kecepatan dan kemudahan (Speed)

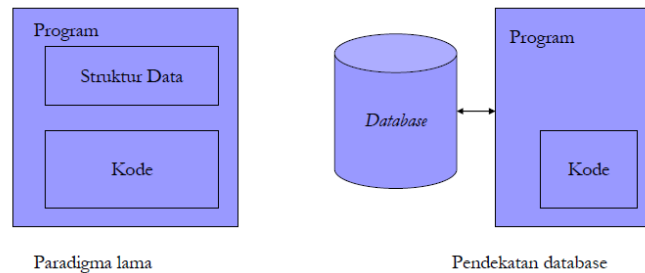
Mesin mampu mengambil dan mengolah informasi jauh lebih cepat dibandingkan dengan manusia.

2. Efisiensi ruang penyimpanan (Space)

Data dikodekan secara elektronik dan disimpan dalam sebuah media (Harddisk).

3. Independensi Program – Data

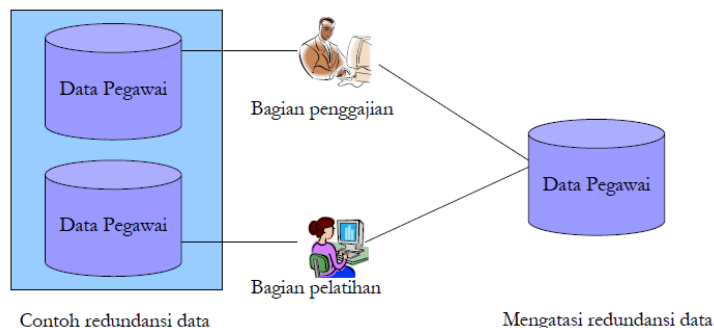
- Struktur data pada basis data terpisah dengan program
- Perubahan struktur data tidak membuat program harus dimodifikasi.



Gambar 4. independensi program - data

4. Meminimalkan redundansi data

- Redundansi data dapat dikurangi dengan cara data yang sama untuk aplikasi yang berbeda dijadikan satu.



Gambar 5. redundansi data

5. Meningkatkan Konsistensi Data

- Pengurangan redundansi data berimplikasi pada peningkatan konsistensi data (mengurangi kemungkinan untuk tidak konsisten).
- Contoh : Dua biro perjalanan tidak terhubung ke *database* (basis data) milik maskapai penerbangan. Apa yang terjadi kalau salah satu biro sudah menjual tempat duduk nomor 4 padahal biro yang lain tidak tahu?

6. Meningkatkan Kemampuan Berbagi data

- Data dapat diakses oleh banyak pemakai dengan tetap memperhatikan otorisasi.

- Istilah *multiuser* menyatakan bahwa sebuah data bisa diakses oleh banyak orang dalam waktu yang bersamaan.
7. Meningkatkan pencapaian standarisasi
 - Standarisasi seperti nama data, panjang data, kemungkinan nilainya, dan bahkan prosedur untuk mengaksesnya dapat diatur oleh yang berwenang (DBA). Contoh : Nama pegawai selalu betipe Alphanumeric dengan panjang maksimal 35 karakter. Semua pemrogram menggunakan standar tersebut.
 8. Meningkatkan kualitas data.
 - Kualitas data sangat berpengaruh terhadap pemerolehan informasi yang berkualitas.
 - Adanya kekangan (*constraint*) dalam *database* membuat pelanggaran terhadap isi data oleh pemakai tidak akan ditoleransi oleh sistem dengan sendirinya.
 - Kekangan adalah suatu aturan yang diterapkan pada data dan tidak bisa dilanggar oleh pemakai.
 - Contoh : Agama hanya bisa diisi dengan I, K, H, B, P. Sistem *database* akan menolak kalau huruf X dicoba untuk dimasukkan.
 9. Mengurangi pemeliharaan program.
 - Perubahan terhadap struktur data dengan berbagai alasan seringkali dilakukan selama tahapan pemeliharaan, misalnya data baru ditambahkan atau panjang suatu data ditambah. Perubahan seperti ini tidak selalu membuat program-program yang telah jadi harus ikut diubah.

Kerugian pemakaian sistem basis data

1. Mahal
 - Diperlukan hardware tambahan : CPU yang lebih besar, Terminal yang lebih banyak, Alat untuk komunikasi.
 - Biaya performance yang lebih besar : Listrik, Personil yang lebih tinggi klasifikasinya, Biaya telekomunikasi yang antar lokasi/kota.
2. Kompleks
3. Kesulitan dalam pemeliharaan (prosedur backup & recovery sulit).

1.5 Penggunaan Aplikasi Basis Data (*Database*)

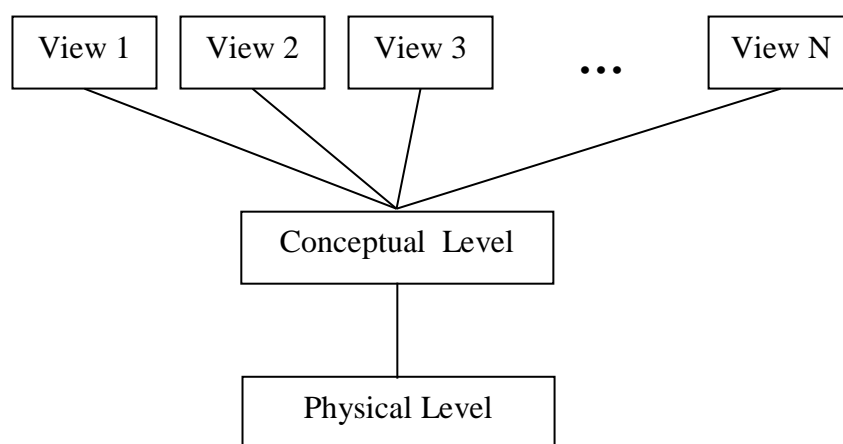
1. Perbankan : pengelolaan data nasabah, akunting, rekening, pinjaman, transaksi perbankan.
2. Penerbangan : pengelolaan data reservasi dan penjadwalan
3. Jasa Paket : tarif paket, tracking paket
4. Lembaga pendidikan (universitas/ perguruan tinggi) : pengelolaan pendaftaran, informasi mahasiswa, registrasi matakuliah, daftar nilai.
5. kepegawaian : pengelolaan data pegawai, riwayat pekerjaan, gaji
6. Telekomunikasi : tariff SLJJ, tagihan bulanan, saldo kartu Prabayar.
7. Toko buku : daftar buku, riwayat pembelian.
8. Penjualan : pengelolaan data customer, produk, penjualan
9. Pabrik : pengelolaan data produksi, persediaan barang, pemesanan, agen

1.6 Arsitektur Basis Data

Sesuai ANSI SPARC 1975, arsitektur untuk sistem basis data terdiri dari tiga tingkat yaitu:

1. Tingkat Internal atau fisik / Physical Level
2. Tingkat Konsep / Conceptual Level
3. Tingkat Pandangan / View Level / External Level

Digambarkan sebagai berikut :



Gambar 6. Arsitektur 3 tingkat

Tujuan adanya arsitektur 3 tingkat adalah memunculkan abstraksi dan memisahkan pandangan pemakai terhadap basis data dan cara implementasinya

secara fisik sehingga memperoleh keuntungan ketidakbergantungan / independensi data yang tinggi.

Alasan :

1. Tiap pemakai dapat mengakses data yang sama tetapi dengan pandangan yang berbeda, dengan kata lain user view sesuai kebutuhan dan bisa berbeda tiap user.
2. User tidak berurusan dengan kerumitan penyimpanan fisik basis data.
3. DBA dapat melakukan perubahan struktur penyimpanan basis data tanpa mempengaruhi user view.

Physical Level / Tingkat Internal / Tingkat Fisik

1. Merupakan tingkat terendah.
2. Mendeskripsikan cara penyimpanan fisik data.
3. Mengurusi alokasi ruang penyimpanan data dan indeks.
4. Berhubungan dengan metode manajemen file untuk menyimpan dan mengakses data.

Conceptual Level / Tingkat Konsep

1. Mendeskripsikan tentang data apa yang disimpan.
2. Menyatakan Entitas, Atribut dan keterhubungannya (relationships) serta konstrain maupun informasi mengenai keamanan dan integritas data.
3. Berisi Struktur Logik.

View Level / Tingkat Pandangan / External Level

1. Berinteraksi dengan sebagian dari basis data (sesuai kebutuhan user).
2. View / pandangan mendefinisikan satu bagian untuk satu kelompok pemakai tertentu.
3. Dapat menyediakan banyak pandangan berbeda pada basis data yang sama.

1.7 Ketakbergantungan / Independensi Data

Kemampuan memodifikasi skema di satu tingkat tidak mempengaruhi definisi skema di tingkat yang lain

Ada 2 macam independensi :

1. Logical data independence / ketidakbergantungan data secara logik
 - a. Perubahan skema konsep tak pengaruhi skema pandangan / program aplikasi.

- b. Misal : penambahan suatu atribut dimungkinkan tanpa harus menulis ulang program aplikasi.
 - c. Catatan : Perubahan skema konsep untuk aplikasi yang berhubungan ada kemungkinan tetap mempengaruhi tingkat pandangan, contohnya penambahan atribut golongan darah pada struktur data mahasiswa → agar dapat diisi dengan mudah maka program aplikasi yang sudah ada juga harus sedikit dirubah untuk fasilitas pengisian gol. Darah.
2. Physical data independence / Ketakbergantungan secara fisik
 - a. Perubahan skema fisik tak pengaruhi skema konsep.
 - b. Misal : Penambahan indeks seharusnya tidak akan mempengaruhi tingkat konsep maupun pandangan, hanya mungkin yang cukup terasa pada tingkat pandangan adalah peningkatan kinerja.

1.8 Model Data

Konsep yang digunakan untuk membuat struktur Basis data terdapat 2 kelompok model data :

1. Konseptual / high level
Bagaimana pemakai basis data memandang / memperlakukan data.
2. Fisikal / Low Level
Bagaimana deskripsi detail data disimpan dalam komputer (format perekaman, urutan dan indexing).

Model data konseptual

Terdiri dari 2 macam :

1. Object base data model (Model data berbasis object)
Menjelaskan hubungan logik antar data dalam suatu basis data berdasar obyek datanya.
Mis : Entity Relationship (E-R), Semantik, Object Oriented
2. Record Base data Model (Model Data berbasis Record)
Menjelaskan hubungan logik antar data berdasar record.
Digunakan untuk menguraikan implementasi sistem basis data.
Mis : Relational, Hirarki, Jaringan.

BAB II

KONSEPTUAL BASIS DATA DAN ENTITAS-RELASIONAL

Tujuan :

1. Mengetahui beberapa teknik alternatif pemodelan data
2. Memahami konsep relasi
3. Mampu menggunakan simbol-simbol ERD

2.1 E-R Model

1. Entity Relationship Model → Dikenalkan oleh Chen (1976).
2. Berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (Entity) serta hubungan (Relationship) antara entitas-entitas tersebut.
3. Tak Bergantung DBMS dan platform perangkat keras.

Digunakan untuk :

1. Mengembangkan Model Konseptual
2. Menjelaskan Struktur Basis Data
3. Memberikan gambaran kepada pengguna terhadap data

Tiga Komponen Penting dalam Model E-R adalah : **Entity, Atribut dan Relasi**

ENTITY / ENTITAS

→ Obyek di dunia nyata yang dapat dibedakan dengan obyek yang lainnya
Set Entity / Himpunan Entitas : Kumpulan Entity yang serupa.

Set Entity dapat berupa :

Fisik : Mahasiswa, Pasien, Kendaraan

Konsep / Logik : Pekerjaan, Mata Kuliah, Kursus

Simbol Entity : (berupa kotak Segiempat)

Contoh Entity → Prabu, Fadil, Afif → dimodelkan sebagai Mahasiswa

ATRIBUT

→ Ciri atau karakteristik yang bermakna untuk mendeskripsikan entitas

→ Bertujuan untuk membedakan obyek-obyek di dalam entity

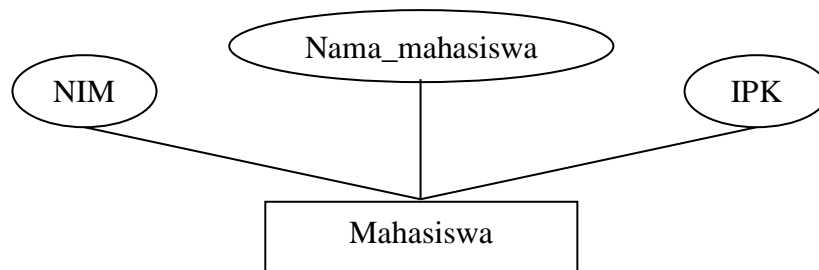
Karakteristik yang bermakna dan tidak bermakna

contoh untuk entity mahasiswa maka :

1. Bermakna : NIM, nama_mahasiswa, IPK
2. Tidak bermakna : tinggi_badan, berat badan

Tinggi_badan dan berat_badan menjadi karakteristik yang bermakna bagi entity petinju.

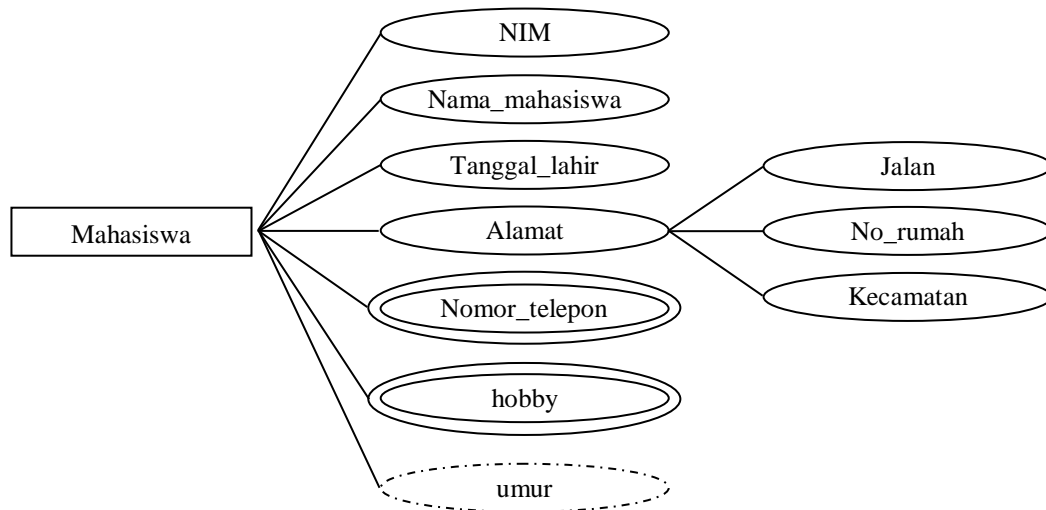
Simbol atribut : elips dan dihubungkan dengan entity melalui garis

**JENIS-JENIS ATRIBUT :**

1. Atribut Sederhana
 - a. Atribut yang disusun hanya dari satu komponen tunggal dengan keberadaan bebas. Atribut sederhana tak dapat dibagi lagi → atribut atomik
 - b. Misal : nama, jenis_kelamin
2. Atribut Komposit
 - c. Atribut yang disusun dari banyak komponen yang masing-masing keberadaannya bebas. Atribut dapat dipecah menjadi komponen yang lebih kecil yang keberadaannya bebas.
 - d. Misal : alamat → jalan, no_rumah, kecamatan
3. Atribut Tunggal
 - a. Mengandung hanya satu nilai. Bisa terdiri dari atribut sederhana atau atribut komposit.
 - b. Misal : NIM, No_karyawan, Agama.
 - c. Ket : Seorang mahasiswa hanya punya 1 NIM, atau 1 Agama.

4. Atribut Bernilai Banyak / Jamak
 - a. Atribut yang mengandung banyak nilai.
 - b. Misal : hobby, ketrampilan, nomor_telepon.
 - c. Ket : Seorang mahasiswa bisa memiliki banyak hobby atau lebih dari 1 nomor_telepon.
5. Atribut turunan
 - a. Atribut yang mengandung nilai dimana nilai tersebut bisa diperoleh dari hasil kalkulasi atau penerapan algoritma ke atribut lain.
 - b. Misal : atribut umur → bisa diperoleh dari atribut tanggal_lahir.

Cara penggambaran beberapa jenis atribut di atas :



Gambar 6. Cara Penggambaran Atribut

Domain Atribut

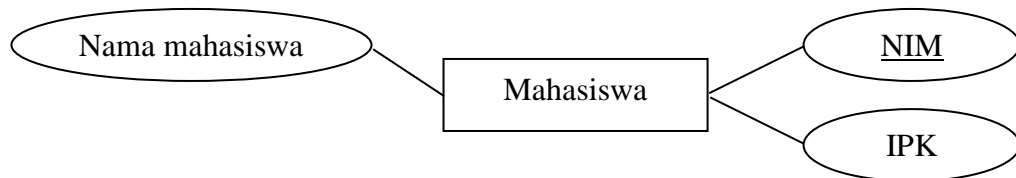
1. Himpunan nilai yang diberikan ke suatu atribut.
2. Termasuk diantaranya : tipe data, panjang, konstrain, format, dll
3. Atribut yang berbeda boleh mempunyai domain yang sama.

Key (Kunci)

1. Himpunan dari satu atau lebih atribut yang digunakan untuk membedakan antara satu entity dengan entity yang lain.
2. Harus unik dan tidak boleh kosong.
3. Superkey Candidate key, primary key.

4. Primary Key :

- Candidate key yang dipilih untuk identifikasi entitas pada himpunan entitas.
- Digambarkan dengan menambah garis bawah.

**RELASI**

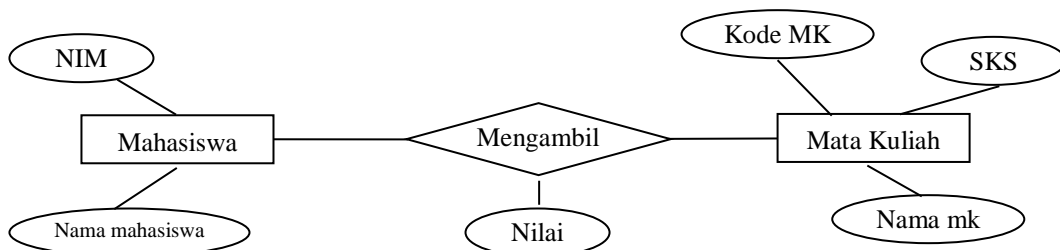
- Himpunan antara satu himpunan entitas dengan himpunan entitas yang lain.
- Dilambangkan dengan simbol diamond (wajik) yang di dalamnya terdapat kata kerja / kata hubung,

Misal :

Dosen Mengajar Mahasiswa

**Atribut Hubungan / atribut relasi**

- Uraian tentang suatu hubungan
- Berguna untuk menjelaskan suatu hubungan
- Jika atribut hubungan cukup banyak maka perlu dipertimbangkan untuk menjadi entity baru.

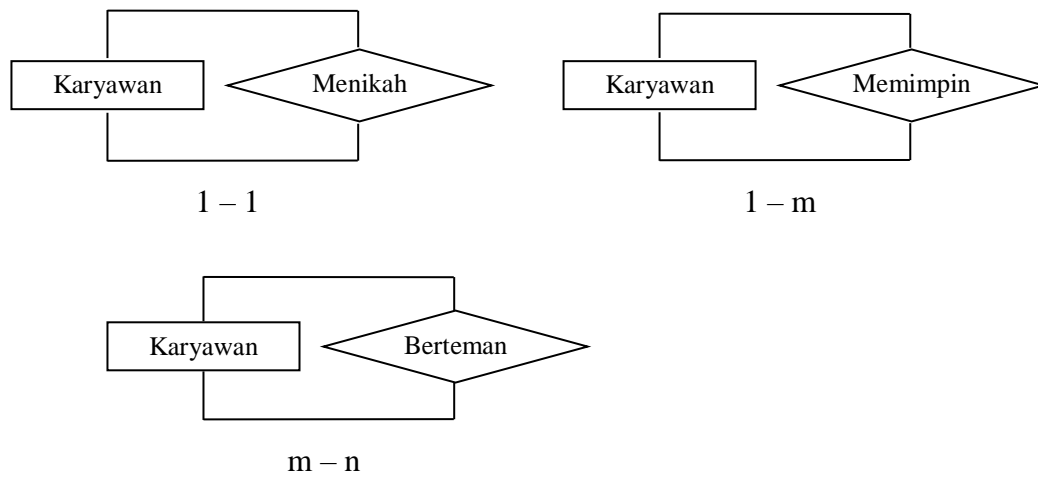
**Derajat Relasi :**

- Jumlah entitas yang berpartisipasi dalam suatu relasi

Ada tiga macam :

- Unary relationship / rekursif relationship

Hanya melibatkan 1 entitas / berderajat 1



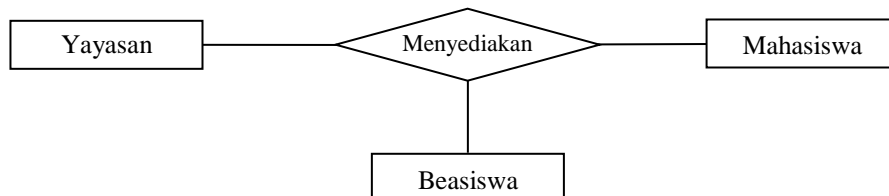
b. Binary relationship

Melibatkan 2 entitas / berderajat 2



c. Ternary Relationship

Melibatkan 3 entitas / berderajat 3



BAB III

TRANSFORMASI DIAGRAM E-R KE RELASI

Tujuan :

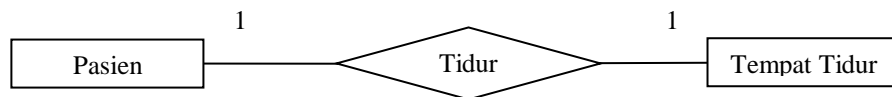
1. Memahami cara mentransformasi diagram ER ke relasi

3.1 Cardinality Ratio

1. Menjelaskan hubungan batasan jumlah keterhubungan satu entity dengan entity lainnya.
2. Menyatakan jumlah anggota entity yang terlibat dalam ikatan yang terjadi.
3. Tergantung pada aturan satu organisasi / aturan bisnis tertentu.
4. Jenis :
 - a. One to one, 1 : 1, satu ke satu
 - b. One to many, 1 : m, satu ke banyak
 - c. Many to many, m : n, banyak ke banyak

Contoh :

1 : 1



Seorang pasien hanya boleh menempati satu tempat tidur

Satu tempat tidur hanya boleh ditempati seorang pasien

Prabu	—	—	A11
Fadil	—	—	B12
Afif	—	—	A12

1 : m, m : 1



Seorang pegawai hanya boleh bekerja di satu departemen

Satu departemen bisa mempekerjakan banyak pegawai

Prabu		Dep. Pertanian
Fadil		Dep. Perindustrian
Afif		Dep. Peternakan

m : n



Seorang mahasiswa boleh mengambil banyak mata kuliah

Satu mata kuliah boleh diambil banyak mahasiswa

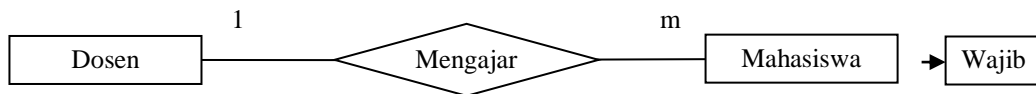
Prabu		Kalkulus
Fadil		Basis Data
Afif		Matematika Diskrit

3.2 Partisipasi Hubungan

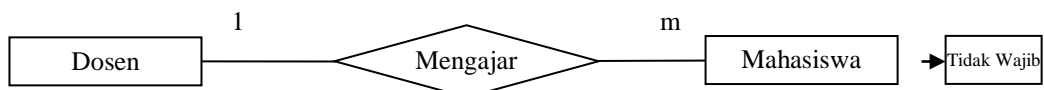
1. Partisipasi atau keterlibatan tiap anggota dalam membentuk hubungan.
2. Dapat bersifat wajib (**Full / Total Participation**) atau tidak wajib (**Partial Participation**)
3. Penggambaran bisa menggunakan (berurutan lambang wajib / tidak wajib)

==== / — atau — / - - - - - atau — □ / — ○ □

Misal :



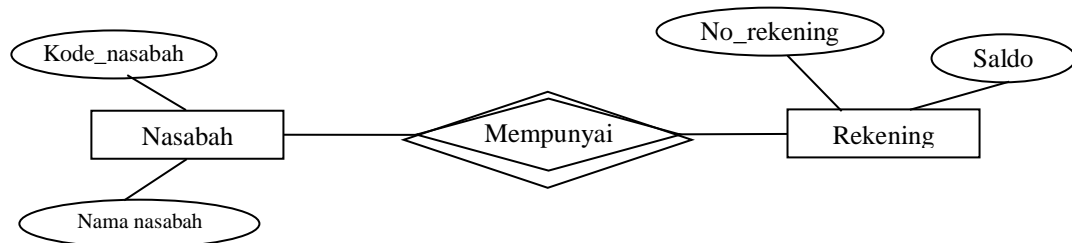
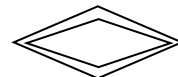
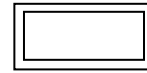
Setiap dosen harus membimbing minimal 1 orang mahasiswa.



Dosen bisa membimbing banyak mahasiswa atau tidak membimbing satu orang pun mahasiswa.

Entitas Kuat (Strong Entitas) dan Entitas Lemah (Weak Entity)

1. Entitas Kuat → keberadaannya tak tergantung keberadaan entitas lain.
2. Entitas Lemah → keberadaannya sangat tergantung keberadaan entitas lain.
3. Identifying owner
 - a. Entitas dimana entitas lemah bergantung.
 - b. Digambarkan dengan segiempat bertumpuk.
4. Identifying relationship
 - a. Relasi antara entitas lemah dan entitas kuat.
 - b. Digambarkan dengan diamond / wajik bertumpuk

**Soal :**

Sebuah Politeknik mempunyai 3 Buah Jurusan (TI, TM, TP). Setiap Mahasiswa hanya diperbolehkan mengambil 1 jurusan. Apabila mahasiswa jurusan TI akan mengambil Jurusan TM maka mahasiswa tersebut harus keluar dari jurusan TI dan pindah ke jurusan TM. Setiap mata kuliah hanya boleh diampu oleh seorang dosen, meskipun mata kuliah tersebut mata kuliah paralel (mata kuliah yang sama tetapi lebih dari 1 kelas). Apabila Dosen mempunyai kesibukan yang cukup padat seperti sedang menyelesaikan S2/S3 ataupun memegang jabatan struktural maka dosen tersebut diperkenankan untuk tidak mengajar mata kuliah apapun. Selain mengampu mata kuliah, setiap dosen wajib membimbing minimal 1 orang mahasiswa.

Jawaban :

