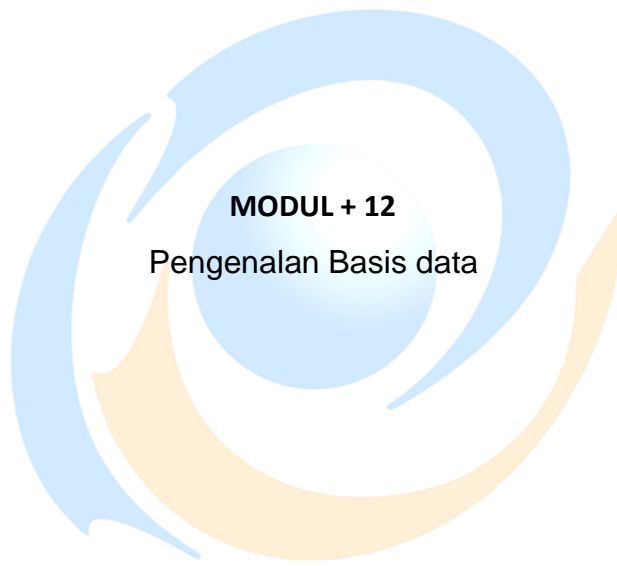


**MODUL + DASAR SISTEM INFORMASI**

**KODE MATAKULIAH :CCE110**



**MODUL + 12**  
Pengenalalan Basis data

Universitas  
**Esa Unggul**

**DISUSUN OLEH**  
**Kundang K Juman**

**UNIVERSITAS ESA UNGGUL**

**2020**

## MODUL 12

### Pengenalan Basis data

*Database-management System* (DBMS) adalah kumpulan data yang saling terkait dan satu set program untuk mengakses datanya. Kumpulan data ini disebut basis data (*database*), yang merupakan kumpulan informasi mengenai fakta-fakta yang disimpan dalam komputer secara sistematis. Tujuan utama DBMS adalah menyimpan dan mencari informasi basis data dengan mudah, cepat, dan efisien. Sistem basis data dirancang untuk mengelola banyak informasi. Data-data ini perlu diolah melalui analisis tertentu sehingga berguna dalam pengambilan keputusan. Basis data sangat erat kaitannya dengan kehidupan sehari-hari, yaitu data perusahaan, data bank, universitas, dan lain-lain. Data tertentu juga dapat diperoleh melalui hasil pengamatan. Pemanfaatan basis data dilakukan dengan tujuan yaitu:

1. kecepatan dan kemudahan (*speed*)
2. efisiensi ruang penyimpanan (*space*)
3. keakuratan (*accuracy*)
4. ketersediaan (*availability*)
5. kelengkapan (*completeness*)

6. keamanan (*security*)

7. data dapat dipakai secara bersama (*shareability*).

Saat ini semua kebutuhan informasi dapat dipenuhi dengan cepat. Mulai mengetahui kurs dolar, jadwal tayang bioskop, banyaknya *subscriber* seorang *you-tuber*, sampai banyaknya *follower* akun seseorang. Selain itu, di era industri 4.0 ini juga mempermudah seseorang dalam melakukan aktifitasnya, yaitu mulai dari memesan tiket, belanja bulanan, membayar uang kos dan lain-lain. Tanpa disadari, semua data informasi ini masuk ke dalam suatu sistem basis data. Sistem ini sangat diperlukan untuk analisis lebih lanjut mengenai semua hal yang akan diteliti. Dengan teknik *data mining*, *server* untuk sistem basis data dapat mengetahui pada tanggal berapa saja jadwal penerbangan padat, kapan waktu yang tepat untuk suatu toko memberi diskon belanja, dan kasus menarik lainnya.

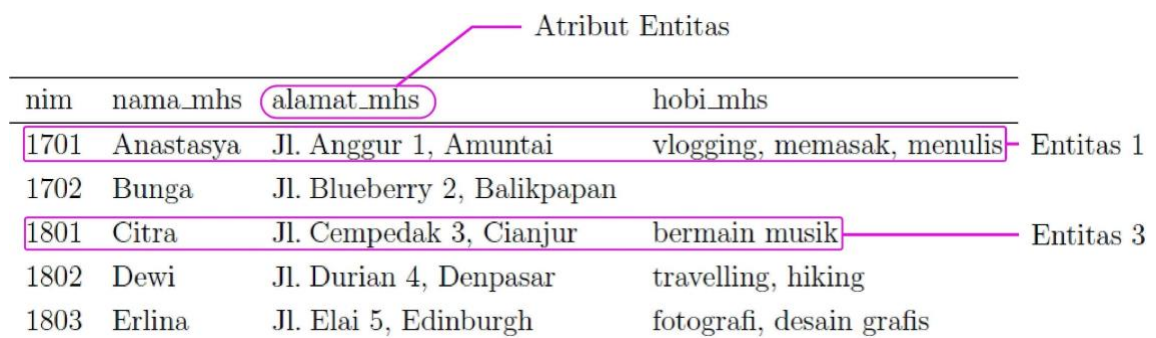
Misalkan di kota Jakarta saat ini sangat viral dengan adanya penjualan roti. Jika pihak penjual membuat suatu aplikasi sehingga pembeli dapat mengakses menu pembelian, maka semua data penjualan yang tersimpan dapat dimanfaatkan oleh penjual untuk meningkatkan pendapatannya. Basis data yang tersimpan dalam *server* dapat berupa waktu pembelian, tanggal pembelian, cuaca saat pembelian, banyaknya pesanan, jenis roti yang dipesan, spesifikasi *handphone* pembeli, dan lain-lain. Bahkan jika pembeli memiliki *member card*, maka penjual roti dapat mengetahui identitas yang lebih banyak lagi untuk digali.

### **Model *Entity-Relationship***

Model *Entity-Relationship* memiliki dua komponen utama pembentuk, yaitu entitas (*entity*) dan relasi (*relation*). Kedua komponen ini dideskripsikan lebih jauh melalui sejumlah atribut atau properti.

## Entitas

Entitas adalah individu yang mewakili sesuatu obyek yang nyata dan dapat dibedakan dari sesuatu yang lain. Contohnya adalah Winda, Ayla, Neni, Nurul, Mobil Toyota, Mobil Honda, Mobil Daihatsu, Melodi, Haruka, Buku aljabar linier, Buku basis data, Nasi kuning, Nasi goreng, Jus wortel, *Milkshake*, Roti balok, dan lain-lain.



nim	nama_mhs	alamat_mhs	hobi_mhs	
1701	Anastasya	Jl. Anggur 1, Amuntai	vlogging, memasak, menulis	Entitas 1
1702	Bunga	Jl. Blueberry 2, Balikpapan		
1801	Citra	Jl. Cempedak 3, Cianjur	bermain musik	Entitas 3
1802	Dewi	Jl. Durian 4, Denpasar	travelling, hiking	
1803	Erlina	Jl. Elai 5, Edinburgh	fotografi, desain grafis	

Gambar 1.1: Himpunan entitas Mahasiswa

Himpunan entitas (*entity set*) adalah sekelompok entitas yang berada dalam lingkup yang sama. Contoh himpunan entitas antara lain Pegawai, Mobil, Pelanggan, Mahasiswa, Buku, Makanan, Minuman, dan lain sebagainya. Himpunan entitas (yang kurang praktis dalam penyebutannya) ini seringkali digantikan dengan sebutan entitas saja. Oleh karena itu, seringkali ditemukan penggunaan istilah entiti-

## Pengenalan Sistem Basis Data

*Database-management System* (DBMS) adalah kumpulan data yang saling terkait dan satu set program untuk mengakses datanya. Kumpulan data ini disebut basis data (*database*), yang merupakan kumpulan informasi mengenai fakta-fakta yang disimpan dalam komputer secara sistematis. Tujuan utama DBMS adalah menyimpan dan mencari informasi basis data dengan mudah, cepat, dan efisien. Sistem basis data dirancang

untuk mengelola banyak informasi. Data-data ini perlu diolah melalui analisis tertentu sehingga berguna dalam pengambilan keputusan. Basis data sangat erat kaitannya dengan kehidupan sehari-hari, yaitu data perusahaan, data bank, universitas, dan lain-lain. Data tertentu juga dapat diperoleh melalui hasil pengamatan. Pemanfaatan basis data dilakukan dengan tujuan yaitu:

1. kecepatan dan kemudahan (*speed*)
2. efisiensi ruang penyimpanan (*space*)
3. keakuratan (*accuracy*)
4. ketersediaan (*availability*)
5. kelengkapan (*completeness*)
6. keamanan (*security*)
7. data dapat dipakai secara bersama (*shareability*).

Saat ini semua kebutuhan informasi dapat dipenuhi dengan cepat. Mulai mengetahui kurs dolar, jadwal tayang bioskop, banyaknya *subscriber* seorang *you-tuber*, sampai banyaknya *follower* akun seseorang. Selain itu, di era industri 4.0 ini juga mempermudah seseorang dalam melakukan aktifitasnya, yaitu mulai dari memesan tiket, belanja bulanan, membayar uang kos dan lain-lain. Tanpa disadari, semua data informasi ini masuk ke dalam suatu sistem basis data. Sistem ini sangat diperlukan untuk analisis lebih lanjut mengenai semua hal yang akan diteliti. Dengan teknik *data mining*, *server* untuk sistem basis data dapat mengetahui pada tanggal berapa saja jadwal penerbangan padat, kapan waktu yang tepat untuk suatu toko memberi diskon belanja, dan kasus menarik lainnya.

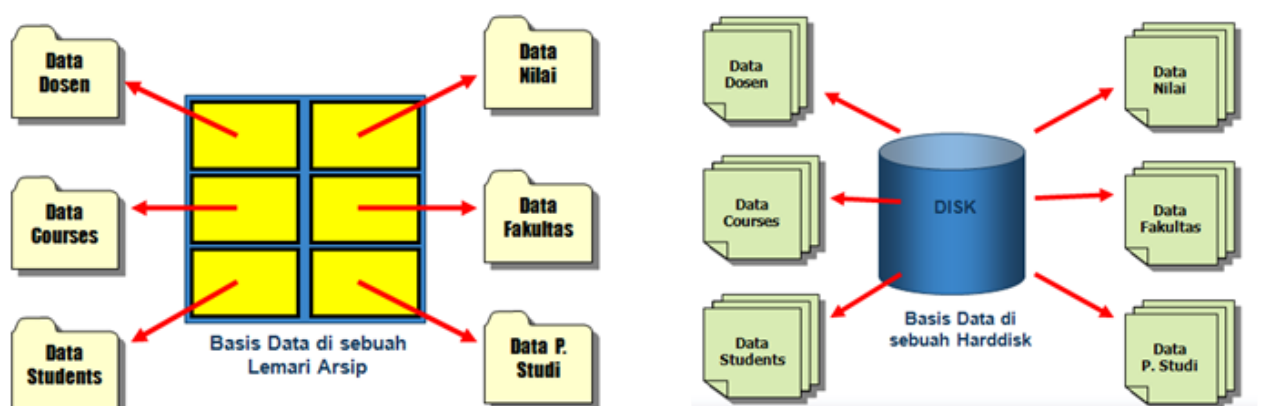
Misalkan di kota Samarinda saat ini sangat viral dengan adanya penjualan roti. Jika pihak penjual membuat suatu aplikasi sehingga pembeli dapat mengakses menu pembelian, maka semua data penjualan yang tersimpan dapat dimanfaatkan oleh penjual untuk meningkatkan pendapatannya. Basis data yang tersimpan dalam *server* dapat berupa waktu pembelian, tanggal pembelian, cuaca saat pembelian, banyaknya pesanan, jenis roti yang dipesan, spesifikasi *handphone* pembeli, dan lain-lain. Bahkan jika pembeli memiliki *member card*, maka penjual roti dapat mengetahui identitas yang lebih banyak lagi untuk digali.

## 1.1 Skala Pengukuran Data

Sebelum mengenal lebih jauh tentang basis data dan penggunaannya, alangkah baiknya jika mempelajari skala pengukuran data terlebih dahulu. Data dibagi menjadi 4 skala, yaitu nominal, ordinal, interval, dan rasio yang dapat dirangkum dalam Tabel 1.1.

Nilai-nilai dari data nominal digunakan ketika data tersebut hanya sebagai pembeda saja. Dalam hal ini, semua data dianggap sama atau setara. Artinya tidak ada datum yang lebih unggul, lebih baik, dan lebih tinggi. Contoh data nominal adalah warna, jenis pekerjaan, jenis kelamin, program studi, dan lain-lain. Lebih mudah dipahami ketika nilai-nilai dari variabel nominal bernilai bilangan bulat. Misalkan warna kesukaan mahasiswa program studi matematika adalah merah marun, biru dongker, hijau toska, kuning langsung, dan putih susu. Peneliti boleh memberi nilai-nilai untuk setiap warna tersebut. Karena data ini bersifat nominal,

maka nilai-nilai warna ini tergantung oleh peneliti. Sebagai contoh, warna merah marun bernilai  $-2$ , biru dongker bernilai  $-1$ , hijau toska bernilai  $0$ , kuning langsung bernilai  $1$ , dan putih susu bernilai  $2$ . Peneliti lain mungkin saja memiliki penilaian berbeda. Artinya nilai data nominal boleh ditukar.



Skala	Tipe data	Operasi	Fitur pembeda	Ukuran pemusatan data
Nominal	diskrit	$=, f=$	hanya kategori	hanya modus
Ordinal	diskrit	$=, f=, \leq, \geq$	kategori terurut	modus dan median
Interval	kontinu	$=, f=, \leq, \geq, +, -$	makna interval	modus, median, dan mean
Rasio	kontinu	$=, f=, \leq, \geq, +, -, \times, \div$	nilai nol mutlak	modus, median, dan mean

Selanjutnya adalah data ordinal. Dalam data ordinal, urutan data diperhatikan. Artinya, urutan nilai dari data ordinal harus monoton. Salah satu contoh data ordinal adalah tingkat kelezatan suatu makanan. Seperti halnya data nominal, Peneliti menginput jika nilai dari data ordinal diubah ke dalam bilangan bulat. Karena "sangat lezat" lebih baik daripada "lezat" dan "lezat" lebih baik daripada "hambar", maka Peneliti boleh menginput tingkat "hambar" bernilai 1, tingkat "lezat" bernilai 2, dan tingkat "sangat lezat" bernilai 3. Peneliti dilarang memberi nilai untuk tingkat "hambar" sebesar 3, jika nilai untuk tingkat "lezat" dan "sangat lezat" masing-masing adalah 1 dan 2.

Selanjutnya, bayangkan jika banyaknya nilai data ordinal sangat banyak lalu dikumpulkan menjadi bilangan yang padat seperti bilangan rasional. Semua nilai-nilai ini memiliki peningkatan yang konstan dan relatif kecil. Data ordinal ini dapat dikatakan data interval. Data interval berbeda dengan istilah selang interval pada matakuliah kalkulus. Data interval lebih menekankan pada sifat dari himpunannya saja. Contoh yang paling mudah untuk mengilustrasikan data interval adalah data suhu (temperatur). Semakin tinggi nilai suhu suatu ruangan, maka keadaan ruangan tersebut semakin panas. Dalam hal ini, suhu  $24^{\circ}\text{C}$  bukan berarti 2 kali lebih panas dari  $12^{\circ}\text{C}$  dan suhu  $12^{\circ}\text{C}$  bukan berarti 2 kali lebih panas dari  $-12^{\circ}\text{C}$ . Suhu ruangan yang tidak panas juga bukan berarti memiliki nilai suhu  $0^{\circ}\text{C}$ .

Skala pengukuran data yang paling tinggi adalah data rasio. Nilai untuk data rasio dapat dikonversi menjadi data interval, data ordinal, atau data nominal.

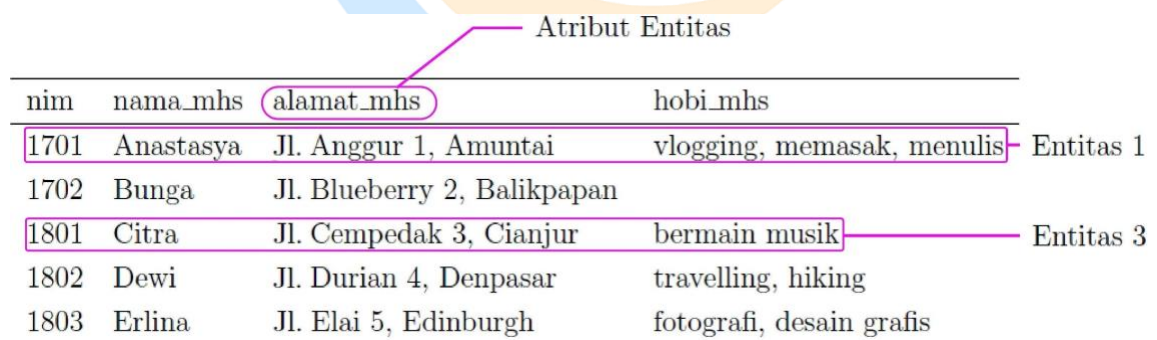
Perhatikan bahwa data interval juga dapat diubah menjadi data ordinal, atau data nominal. Hasil pengukuran data rasio yang bisa dibedakan, diurutkan, memiliki jarak tertentu, dan bisa dibandingkan. Nilai nol mutlak pada data rasio berarti benar-benar menyatakan tidak ada.

## Model *Entity-Relationship*

Model *Entity-Relationship* memiliki dua komponen utama pembentuk, yaitu entitas (*entity*) dan relasi (*relation*). Kedua komponen ini dideskripsikan lebih jauh melalui sejumlah atribut atau properti.

### Entitas

Entitas adalah individu yang mewakili sesuatu obyek yang nyata dan dapat dibedakan dari sesuatu yang lain. Contohnya adalah Winda, Ayla, Neni, Nurul, Mobil Toyota, Mobil Honda, Mobil Daihatsu, Melodi, Haruka, Buku aljabar linier, Buku basis data, Nasi kuning, Nasi goreng, Jus wortel, *Milkshake*, Roti balok, dan lain-lain.



Atribut Entitas

nim	nama_mhs	alamat_mhs	hobi_mhs	
1701	Anastasya	Jl. Anggur 1, Amuntai	vlogging, memasak, menulis	Entitas 1
1702	Bunga	Jl. Blueberry 2, Balikpapan		
1801	Citra	Jl. Cempedak 3, Cianjur	bermain musik	Entitas 3
1802	Dewi	Jl. Durian 4, Denpasar	travelling, hiking	
1803	Erlina	Jl. Elai 5, Edinburgh	fotografi, desain grafis	

Gambar 1.1: Himpunan entitas Mahasiswa

Himpunan entitas (*entity set*) adalah sekelompok entitas yang berada dalam lingkup yang sama. Contoh himpunan entitas antara lain Pegawai, Mobil, Pelanggan, Mahasiswa, Buku, Makanan, Minuman, dan lain sebagainya. Himpunan entitas (yang kurang praktis dalam penyebutannya) ini seringkali digantikan dengan sebutan entitas saja. Oleh karena itu, sering kali ditemukan penggunaan istilah enti-



tas yang berarti himpunan entitas. Dalam sistem basis data, istilah entitas merujuk pada nama tabel. Karena penggunaan entitas sebagai nama tabel, maka baris ke *i* dari entitas disebut data ke *i*. Perhatikan bahwa Gambar 1.1 merupakan entitas Mahasiswa.

## Atribut

Komponen selanjutnya adalah atribut. Komponen atribut adalah karakteristik (properti) yang mendeskripsikan suatu entitas. Dalam sistem basis data, atribut ini merujuk pada kolom tabel. Sebagai contoh, atribut pada entitas Mahasiswa yang ditampilkan dalam Gambar 1.1 adalah *nim*, *nama mhs*, *alamat mhs*, dan *hobi mhs*.

Atribut dibagi menjadi tujuh jenis yaitu atribut sederhana, atribut komposit, atribut bernilai tunggal, atribut bernilai banyak, atribut harus bernilai (*mandatory attribute*), atribut nilai *null*, dan atribut turunan. Beberapa atribut ditunjukkan dalam Gambar 1.2. Adapun penjelasan masing-masing jenis ini adalah sebagai berikut:

1. Atribut sederhana adalah atribut atomik yang tidak dapat dipilah lagi. Contohnya adalah jenis kelamin dan jurusan.
2. Atribut komposit adalah atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna. Sebagai contoh, pada atribut Nama dapat diuraikan menjadi nama depan, nama tengah, nama belakang. Contoh lainnya adalah atribut alamat (nama jalan, nomor rumah, kota).
3. Atribut bernilai tunggal adalah atribut yang hanya memiliki maksimal satu nilai di tiap datanya. Contohnya NIM, nama, dan tanggal lahir.
4. Atribut bernilai banyak adalah atribut yang dapat diisi lebih dari satu nilai. Misalnya *hobby* dan nomor *handphone*.
5. Atribut harus bernilai (*mandatory attribute*) adalah atribut yang harus berisi suatu data. Contohnya adalah NIM, nama mahasiswa, dan alamat.
6. Atribut nilai *null* adalah atribut yang belum memiliki nilai. Dalam hal ini, *null* artinya kosong. Contohnya, pada atribut hobi mhs untuk entitas mahasiswa

(Gambar 1.2) bernama Bunga yang memang tidak memiliki *hobby*. Atribut *null* juga dapat diartikan bahwa datanya belum siap atau masih meragukan.

- Atribut turunan adalah atribut yang nilainya dapat diturunkan dari atribut lainnya. Atribut turunan sebenarnya hanya digunakan sebagai penjelas dan dapat ditiadakan. Contoh atribut turunan adalah angkatan. Nilai-nilai pada atribut angkatan dapat diketahui dari atribut nim (biasanya dua karakter pertama dalam nim menyatakan dua digit bilangan tahun masuknya mahasiswa yang bersangkutan). Contoh atribut turunan lainnya adalah atribut indeks prestasi yang dapat diperoleh dari hasil perhitungan nilai.

Mandatory Attribute			Non Mandatory Attribute
Atribut bernilai tunggal			Atribut bernilai banyak
nim	nama_mhs	alamat_mhs	hobi_mhs
1701	Anastasya	Jl. Anggur 1, Amuntai	vlogging, memasak, menulis
1702	Bunga	Jl. Blueberry 2, Balikpapan	
1801	Citra	Jl. Cempedak 3, Cianjur	bermain musik
1802	Dewi	Jl. Durian 4, Denpasar	travelling, hiking
1803	Erlina	Jl. Elai 5, Edinburgh	fotografi, desain grafis

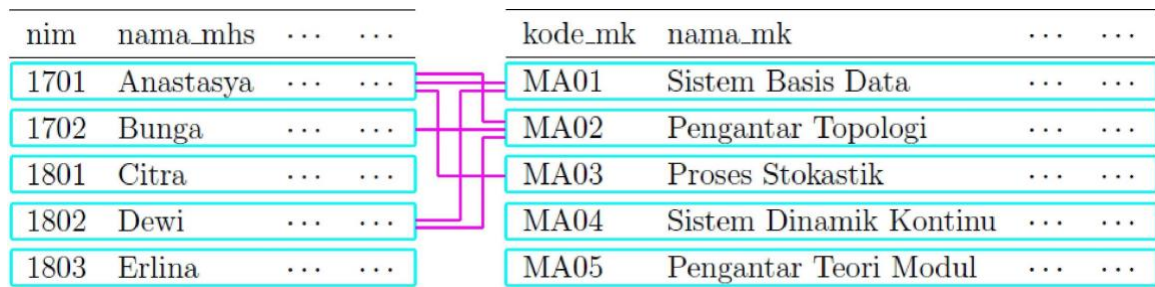
Gambar 1.2: Beberapa jenis atribut

**Primary key** adalah atribut yang dapat membedakan data satu dengan data yang lainnya dalam suatu entitas. Contohnya adalah NIM, Kode makanan, dan Kode barang. Fungsi penggunaan **Primary Key** adalah untuk membedakan data satu dengan data yang lainnya.

## Relasi dan himpunan relasi

Beberapa entitas yang berbeda dapat saling berkaitan dan dapat dihubungkan. Dalam sistem basis data, hubungan ini dinamakan relasi (**relationship**). Jadi, relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Contohnya adalah transaksi barang, Kartu Rencana Studi (KRS), Pembelian, dan lain sebagainya. Perhatikan Gambar 1.3, bahwa ma-

hasiswa dengan nim=' 1702' dan nama mhs=' Bunga' (dalam himpunan entitas Mahasiswa) memiliki relasi dengan entitas mata kuliah dengan kode mk=' MA02' dan nama mk=' Pengantar Topologi' . Relasi di antara kedua entitas tadi mengandung arti bahwa mahasiswa bernama Bunga sedang mempelajari mata kuliah Pengantar Topologi di sebuah program studi.



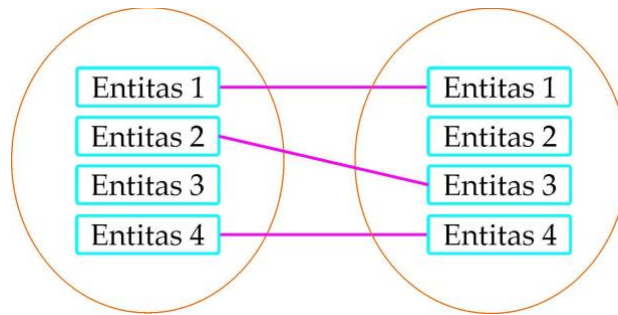
Gambar 1.3: Contoh relasi antara himpunan entitas Mahasiswa dan himpunan entitas mata kuliah

Himpunan relasi (*relationship sets*) merupakan kumpulan semua relasi dalam suatu sistem basis data. Seperti halnya penggunaan istilah himpunan entitas yang disingkat menjadi entitas (walaupun sebenarnya memiliki perbedaan makna), istilah himpunan relasi jarang sekali digunakan dan lebih sering disingkat dengan istilah relasi saja.

Perhatikan bahwa relasi antar entitas pada Gambar 1.3 hanya berlaku pada semester tertentu saja. Kelak akan ditunjukkan adanya faktor waktu dalam relasi antar entitas tersebut yang dapat diterapkan. Semua relasi yang ada di antara himpunan entitas tersebut membentuk sebuah himpunan relasi. Untuk menjelaskan apa yang terjadi di antara kedua himpunan relasi tersebut, maka dapat diberi nama himpunan relasi 'mempelajari' atau himpunan relasi 'belajar'.

Kardinalitas atau derajat relasi adalah jumlah maksimum entitas yang dapat berelasi pada entitas lainnya. Jika diberikan dua himpunan entitas (yaitu **A** dan **B**), maka kardinalitas relasinya dibedakan menjadi 4 jenis, yaitu:

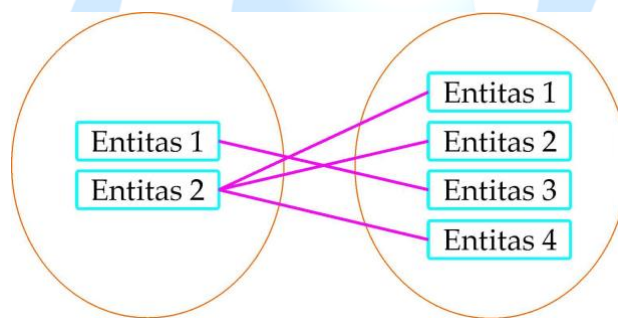
8. Satu ke Satu (*One to One*), yang berarti satu entitas **A** berhubungan paling banyak satu dengan entitas **B**. Begitupun sebaliknya, satu entitas pada himpunan **B** berhubungan paling banyak satu dengan entitas **A**. Perhatikan Gambar 1.4.



Gambar 1.4: Kardinalitas relasi Satu ke Satu

9. Satu ke Banyak (*One to Many*), yang berarti satu entitas *A* dapat berhubungan lebih dari satu dengan entitas *B*. Namun tidak berlaku sebaliknya, dimana setiap entitas pada himpunan entitas *B* berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas *A*. Perhatikan Gambar 1.5.

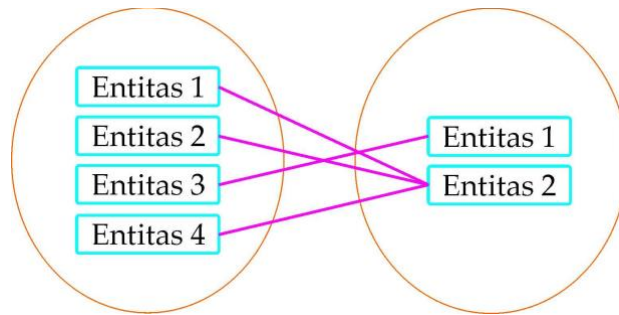
Gambar 1.5: Kardinalitas relasi Satu ke Banyak



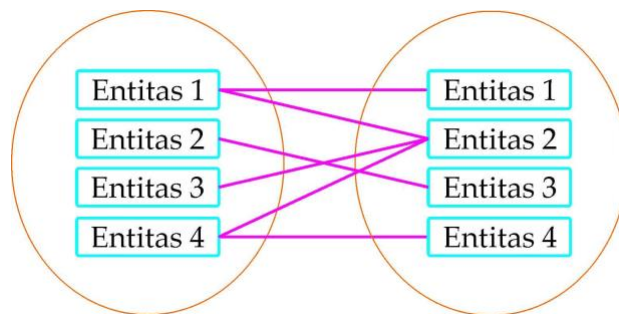
10. Banyak ke Satu (*Many to One*), yang berarti satu entitas *A* berhubungan paling banyak dengan satu entitas *B*. Namun tidak berlaku sebaliknya, dimana setiap entitas pada himpunan entitas *B* dapat berhubungan lebih dari satu dengan entitas *A*. Perhatikan Gambar 1.6.

Kardinalitas relasi Satu ke Banyak dan Kardinalitas relasi Banyak ke Satu dapat dianggap sama, karena tinjauan kardinalitas relasi selalu dapat dilihat dari dua sisi. Artinya, posisi himpunan entitas *A* dan himpunan entitas *B* dapat saja ditukar.

11. Banyak ke Banyak (*Many to Many*), yang berarti satu entitas *A* dapat berhubungan lebih dari satu dengan entitas *B* dan sebaliknya. Perhatikan Gambar 1.7.



Gambar 1.6: Kardinalitas relasi Banyak ke Satu



Gambar 1.7: Kardinalitas relasi Banyak ke Banyak

Contoh kasus untuk relasi Satu ke Satu adalah satu dosen mengepalai satu jurusan di suatu universitas. Dalam hal ini seorang dosen paling banyak mengepalai satu jurusan, walaupun ada dosen yang tidak mengepalai jurusan manapun. Sebaliknya, setiap jurusan pasti dikepalai oleh paling banyak satu dosen. Selanjutnya diagram seperti pada Gambar 1.4 untuk kasus ini dapat disederhanakan menjadi Gambar 1.8.



Gambar 1.8: Penyederhanaan relasi Satu ke Satu

Contoh kasus untuk relasi Satu ke Banyak adalah satu dosen dapat mengajar banyak mata kuliah. Selanjutnya diagram seperti pada Gambar 1.5 untuk kasus ini dapat disederhanakan menjadi Gambar 1.9.



Gambar 1.9: Penyederhanaan relasi Satu ke Banyak

Contoh kasus untuk relasi Banyak ke Banyak adalah satu mahasiswa mempelajari

banyak mata kuliah, dan sebaliknya satu mata kuliah juga dipelajari oleh banyak mahasiswa. Selanjutnya diagram seperti pada Gambar 1.7 untuk kasus ini dapat disederhanakan menjadi Gambar 1.10.



Gambar 1.10: Penyederhanaan relasi Banyak ke Banyak

### *Entity-Relationship Diagram (ERD)*

Model **Entity-Relationship** yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut, dapat digambarkan dengan lebih sistematis menggunakan Diagram **Entity-Relationship** atau **Entity-Relationship Diagram (ERD)**. Notasi simbolik di dalam ERD yang wajib diketahui adalah sebagai berikut:

1. Persegi panjang, menyatakan himpunan entitas.
2. Elips, menyatakan atribut. Khusus untuk atribut yang berfungsi sebagai *primary key*, harus digarisbawahi.
3. Belah ketupat, menyatakan himpunan relasi.
4. Garis, sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.
5. Kardinalitas relasi.

Berikut adalah contoh penggambaran relasi antar himpunan entitas beserta kardinalitas relasi dan atribut-atributnya:

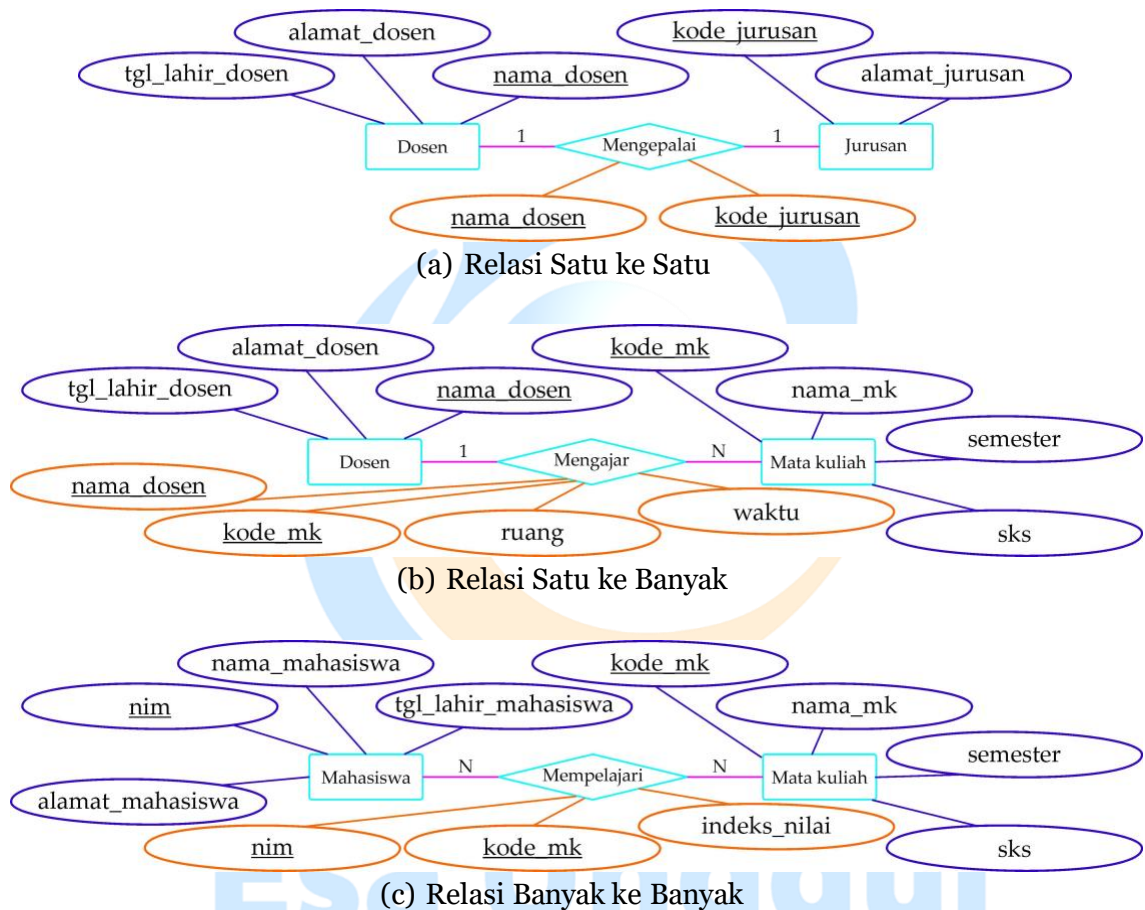
1. Relasi Satu ke Satu (*one-to-one*)  
Contoh ERD untuk relasi pada Gambar 1.8 adalah seperti pada Gambar 1.11a.
2. Relasi Satu ke Banyak (*one-to-many*)  
Contoh ERD untuk relasi pada Gambar 1.9 adalah seperti pada Gambar 1.11b.



### 3. Relasi Banyak ke Banyak (*many-to-many*)

Contoh ERD untuk relasi pada Gambar 1.10 adalah seperti pada Gambar 1.11c.

Penggabungan semua relasi untuk Gambar 1.11a, Gambar 1.11b, dan Gambar 1.11c ditunjukkan pada Lampiran 1.



Gambar 1.11: Contoh ERD

Berikut adalah tahapan dalam perancangan sistem basis data:

1. Mempelajari studi kasus.
2. Membentuk himpunan entitas, himpunan relasi, dan atribut-atributnya.
3. Membuat ERD.
4. Membuat program.



Universitas  
**Esa Unggul**