

# Pertemuan 9

## Perancangan Basis Data (ERD)

### A. Tujuan

1. Mahasiswa memahami makna entitas, atribut, dan relasi.
2. Mahasiswa memahami tahap-tahap normalisasi.
3. Mahasiswa mampu merancang basis data melalui tahap-tahap perancangannya.

### B. Landasan Teori

#### 1. Entitas

*Entity* adalah suatu obyek yang dapat dikenali dari obyek yang lain. Contoh : seseorang yang khusus, perusahaan, peristiwa, tanaman dan lain-lain. *Entity* mempunyai atribut, contoh : seseorang mempunyai nama dan alamat. Kumpulan *entity* adalah suatu kumpulan *entity* dengan tipe yang sama yang berbagi properti yang sama pula. Contoh : kumpulan orang, perusahaan, tanaman, tempat liburan dan lain-lain.

#### 2. Atribut

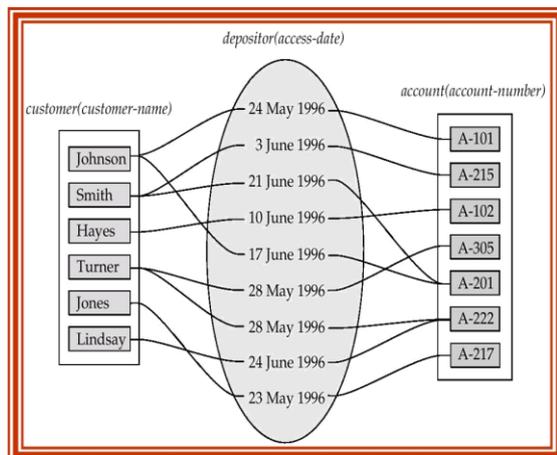
*Entity* ditampilkan oleh sekumpulan *attribute*, yang mana properti deskriptifnya dikuasai oleh seluruh anggota dalam kumpulan *entity*.

Tipe *attribute* :

1. *Simple* (sederhana) dan *composite* (gabungan) *attributes*.
2. *Single-valued* (satu-fungsi ) dan *multi-valued* (multi-fungsi ) *attributes*. Contoh : atribut multi-fungsi : nomor telepon
3. *Derived* (turunan) *attributes* : dapat diperhitungkan dari atribut lain Contoh : umur, tanggal kelahiran.

#### 3. Relasi

*Relationship* adalah kesesuaian antar beberapa *entity*. *Relationship set* adalah hubungan matematika antara *entity*  $n > 2$ , tiap bagiannya diambil dari satuan *entity*  $\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$  dimana  $(e_1, e_2, \dots, e_n)$  adalah *relationship*. *Attribute* dapat merupakan properti dari *relationship set*.



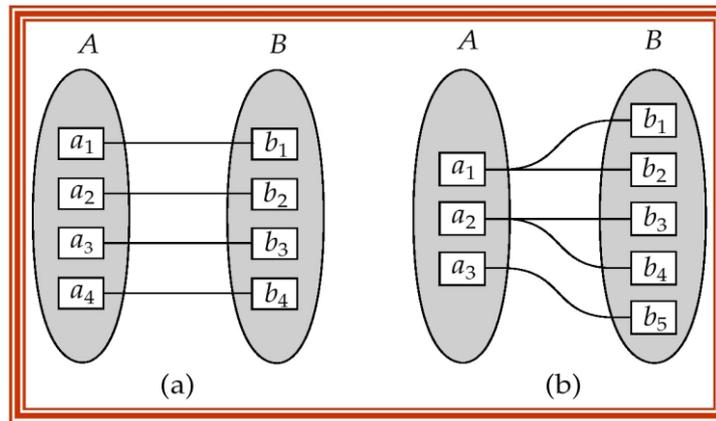
Sebagai contoh: *depositor* merupakan *relationship set* antara *entity sets* *customer* dan *account* mungkin beratribut *access-date*. Mengacu pada jumlah *entity sets* yang terlibat dalam *relationship set*. *Relationship sets* yang melibatkan dua *entity sets* adalah *binary* (atau tingkat dua). Umumnya, hampir semua *relationship sets* dalam sistem database adalah *binary*. *Relationship sets* mungkin melibatkan lebih dari dua *entity sets*.

Contoh : misal seorang pegawai bank bekerja (bertanggung jawab) dalam beberapa cabang, dengan tugas yang berlainan dalam setiap cabang yang berlainan pula. Maka disini terdapat *relationship set ternary* antara *entity sets* pegawai (*employee*), tugas (*job*) dan cabang (*branch*). *Relationships* antara lebih dari dua *entity sets* sangat jarang.

#### 4. Mapping Cardinalities

Mengungkap jumlah entitas ke entitas yang lain bisa dihubungkan melalui *relationship set*. Cardinalitas pemetaan paling banyak digunakan dalam menggambarkan *relationship sets* biner. Untuk *relationship set* biner cardinalitas pemetaan harus merupakan salah satu dari tipe berikut:

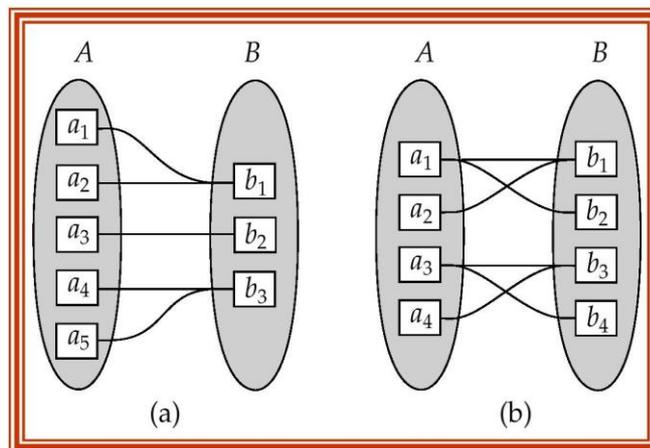
1. *One to one* (satu ke satu)
2. *One to many* (satu ke banyak)
3. *Many to one* (banyak ke satu)
4. *Many to many* (banyak ke banyak)



*One to one*

*One to many*

Catatan : Beberapa elemen di A dan B mungkin tidak dipetakan ke elemen manapun di himpunan lain.



*Many to one*

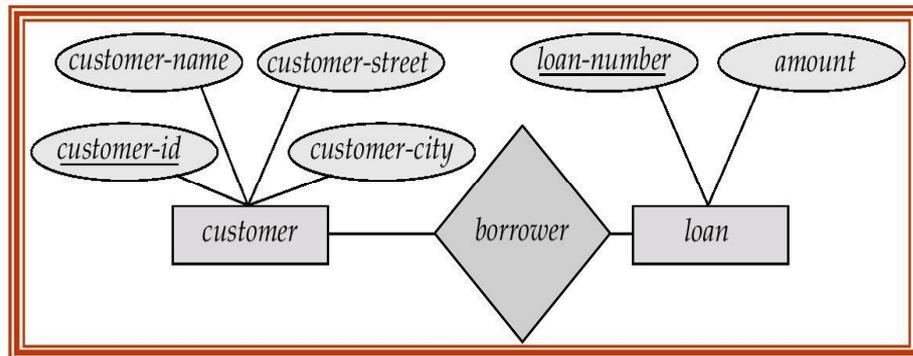
*Many to many*

Catatan : Beberapa elemen di A dan B mungkin tidak dipetakan pada elemen mana pun di himpunan lain. *Mapping Cardinalities* mempengaruhi *ER Design*

## 5. Entity Relationship Diagram

Mencerminkan model database: hubungan antara entities (tabel-tabel) dan relationships (hubungan-hubungan) di antara entities tersebut.

Contoh ERD:



- **Rectangles** melambangkan set-set entitas .
- **Diamonds** melambangkan set-set hubungan (*relationship*) .
- **Lines** menghubungkan atribut dengan set-set entitas serta set-set entitas dengan set-set hubungan (*relationship*).
- **Ellipses** mewakili *attributes*
  - **Double ellipses** mewakili atribut *multivalued* .
  - **Dashed ellipses** menunjukkan sifat yang didapat..
  - **Underline** menunjukkan sifat kunci pokok (*Primary key*)

#### 6. Aturan untuk Rancangan Database

1. Tiap baris harus berdiri sendiri (*independent*): tidak tergantung baris-baris yang lain, dan urutan baris tidak mempengaruhi model database.
2. Tiap baris harus unik: tidak boleh ada 2 atau lebih baris yang sama persis.
3. Kolom harus berdiri sendiri (*independent*): tidak tergantung kolom-kolom yang lain, dan urutan kolom tidak mempengaruhi model database.
4. Nilai tiap kolom harus berupa satu kesatuan: tidak berupa sebuah daftar.

#### 7. Tahap Pembuatan Database

##### a. Tahap 1 : Tentukan Entitas

Tentukan entities (object-object dasar) yang perlu ada di *database*.

Sifat-sifat *entity*:

- Signifikan: memang perlu disimpan di *database*
- Umum: tidak menunjuk pada sesuatu yang khusus
- Fundamental: dapat berdiri sendiri sebagai *entity* yang dasar dan independent
- Unitary: merupakan satu kesatuan yang tidak dapat dipecah lagi

b. Tahap 2 : Tentukan Atribut

Tentukan attributes (sifat-sifat) masing-masing *entity* sesuai kebutuhan database :

- Tentukan sifat-sifat (*fields* atau kolom) yang dimiliki tiap *entity*, serta tipe datanya.
- *Attribute* yang sesuai harus:
  1. Signifikan: memang penting dan perlu dicatat di dalam database
  2. Bersifat langsung (*direct*), bukan *derived*. Contoh *attribute direct*: tanggal\_lahir. Contoh *attribute derived*: umur
- Tentukan *attribute* yang menjadi *Primary Key* untuk *entity* yang bersangkutan.
- Jika satu *attribute* tidak cukup, gabungan beberapa *attribute* bisa menjadi *Composite Primary Key*.
- Jika *Composite Primary Key* banyak (lebih dari 3 *attribute*), sebaiknya menambahkan *attribute* buatan yang menjadi *Primary Key* yang tunggal.

c. Tahap 3 : Tentukan Relasi

Tentukan relationships (hubungan-hubungan) di antara entities tersebut :

- Tentukan jenis hubungan di antara *entity* yang satu dengan entities yang lain.
- Macam hubungan ada 3:
  1. *One-to-one* (1:1)
  2. *One-to-many* (1:n)
  3. *Many-to-many* (m:n)
- Dalam membentuk hubungan di antara 2 entities, tentukan *attribute* mana yang digunakan untuk menghubungkan kedua entities tersebut.
- Tentukan *entity* mana yang menjadi tabel utama, dan *entity* mana yang menjadi tabel kedua.
- *Attribute* (dari tabel utama) yang menghubungkannya dengan tabel kedua menjadi *Foreign Key* di tabel kedua.

d. Tahap 4 : Pembuatan ERD

- Buat *Entity Relationship Diagram* (ERD) berdasarkan hasil dari Tahap 1- 3.
- Ada berbagai macam notasi untuk pembuatan ERD.
- Anda bisa menggunakan *software* khusus untuk menggambar ERD.

e. Tahap 5 : Normalisasi Basis Data

f. Tahap 6 : Implementasi Basis Data

### C. Alat dan Bahan

1. Perangkat komputer dengan OS Windows 7.
2. Modul Praktikum Sistem Basis Data

### D. Langkah-langkah Praktikum

Rancanglah basis data untuk menyelesaikan permasalahan berikut ini.

Suatu perusahaan software diminta membuat basis data yang akan menangani data-data perbankan. Data-data yang akan ditanganinya adalah: data pribadi mengenai nasabah, data account deposit yang dimiliki oleh nasabah, cabang bank di mana nasabah membuka depositnya, dan data transaksi yang dilakukan nasabah. Nasabah boleh mempunyai lebih dari satu account deposit, dan satu account deposit boleh dimiliki oleh lebih dari satu nasabah sekaligus (*joint account*).

Langkah-langkah perancangan database perbankan :

1. Menentukan entities (object-object dasar) yang perlu ada di database.
  - **nasabah**: menyimpan semua data pribadi semua nasabah
  - **rekening**: menyimpan informasi semua rekening yang telah dibuka
  - **cabang\_bank**: menyimpan informasi tentang semua cabang bank
  - **transaksi**: menyimpan informasi tentang semua transaksi yang telah terjadi
2. Menentukan attributes (sifat-sifat) masing-masing *entity* sesuai kebutuhan database
  - **nasabah**:
    - id\_nasabah: nomor id untuk nasabah (integer) PK
    - nama\_nasabah: nama lengkap nasabah (varchar(45))
    - alamat\_nasabah: alamat lengkap nasabah (varchar(255))
  - **rekening**:
    - no\_rekening: nomor rekening (integer) PK
    - pin : personal identification number (varchar(10))
    - saldo: jumlah saldo rekening dalam Rp (integer)
  - **cabang\_bank**:
    - kode\_cabang: kode untuk cabang bank (varchar(10)) PK
    - nama\_cabang: nama lengkap cabang bank (varchar(20))
    - alamat\_cabang: alamat lengkap cabang bank (varchar(255))
  - **transaksi**:
    - no\_transaksi: nomor transaksi (integer) PK
    - jenis\_transaksi: kredit atau debit (varchar(10))
    - tanggal: tanggal terjadinya transaksi (date)
    - jumlah: besarnya transaksi dalam Rp (integer)

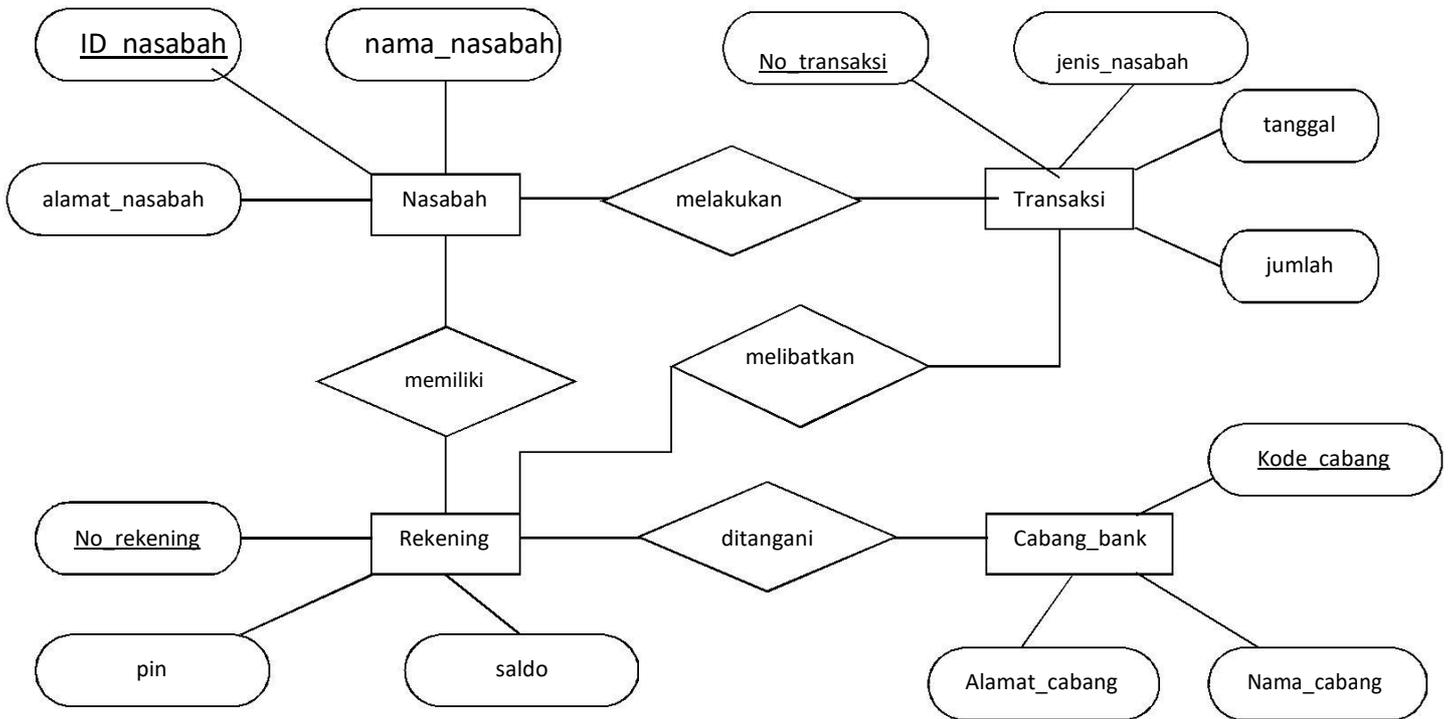
3. Menentukan relationship (hubungan) antar entitas

	<b>nasabah</b>	<b>rekening</b>	<b>cabang_bank</b>	<b>transaksi</b>
<b>nasabah</b>	-	m:n	-	1:n
<b>rekening</b>		-	n:1	1:n
<b>cabang_bank</b>			-	-
<b>transaksi</b>				-

**Hubungan**

- **nasabah** memiliki **rekening**:
  - Tabel utama: **nasabah, rekening**
  - Tabel kedua: **nasabah\_has\_rekening**
  - Relationship: Many-to-many (m:n)
  - Attribute penghubung: **id\_nasabah, no\_rekening** (FK **id\_nasabah, no\_rekening** di **nasabah\_has\_rekening**)
- **nasabah** melakukan **transaksi**:
  - Tabel utama: **nasabah**
  - Tabel kedua: **transaksi**
  - Relationship: One-to-many (1:n)
  - Attribute penghubung: **id\_nasabah** (FK **id\_nasabah** di **transaksi**)
- **cabang\_bank** menangani **rekening**:
  - Tabel utama: **cabang\_bank**
  - Tabel kedua: **rekening**
  - Relationship: One-to-many (1:n)
  - Attribute penghubung: **kode\_cabang** (FK **kode\_cabang** di **rekening**)
- **rekening** terlibat dalam **transaksi**:
  - Tabel utama: **rekening**
  - Tabel kedua: **transaksi**
  - Relationship: One-to-many (1:n)
  - Attribute penghubung: **no\_rekening** (FK **no\_rekening** di **transaksi**)

#### 4. Menggambar ERD Diagram



#### E. Tugas

1. Buatlah rancangan sebuah database untuk menangani data-data kuliah. Data-data yang akan ditanganinya adalah: data pribadi mengenai mahasiswa, data pribadi mengenai dosen, data mata kuliah dan data ruang kelas. Mahasiswa boleh mengambil lebih dari satu mata kuliah, dan satu mata kuliah boleh diambil oleh lebih dari satu mahasiswa sekaligus (*joint account*). Buatlah ER Diagram manual untuk kasus tersebut dari tahap 1 sampai tahap 4!
2. Ambil contoh sembarang database (harus berbeda untuk setiap mahasiswa). Buatlah rancangan ER Diagram manual database tersebut dari tahap 1 sampai tahap 4, dengan ketentuan database minimal mengandung 4 buah entitas.

