

# **PRAKTIKUM SISTEM BASIS DATA**

Nama : Annisa Nugraheni

NIM : L200180066

Kelas : C

## **MODUL 2**

### **PERANCANGAN BASIS DATA**

#### **Tugas**

1. Buatlah rancangan sebuah database untuk menangani data-data kuliah. Data-data yang akan ditangani 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!

#### **Jawab :**

- 1) Menentukan entities (object-object dasar) yang perlu ada di database.

- **Mahasiswa** : menyimpan semua data pribadi mahasiswa
- **Dosen** : menyimpan semua data pribadi dosen
- **Mata\_kuliah** : menyimpan semua data mata kuliah yang di ampu dosen dan yang diambil mahasiswa
- **Ruang\_kelas** : menyimpan semua data ruang kelas untuk setiap mata kuliah

- 2) Menentukan attribute (sifat-sifat) masing-masing entity sesuai kebutuhan database

#### ➤ **Mahasiswa**

- **Id\_mahasiswa** : nomor id untuk mahasiswa (integer)
- **Nama\_mahasiswa** : nama lengkap mahasiswa (varchar(45))
- **Alamat\_mahasiswa** : alamat lengkap mahasiswa (varchar(255))
- **Ttl\_mahasiswa** : tempat tanggal lahir mahasiswa (varchar(50))

#### ➤ **Dosen**

- **Id\_dosen** : nomor id untuk dosen (integer)
- **Nama\_dosen** : nama lengkap dosen (varchar(45))

- **Alamat\_dosen** : alamat lengkap dosen (varchar(255))
- **Ttl\_dosen** : tempat tanggal lahir dosen (varchar(50))
- **Kode\_dosen** : kode untuk dosen (varchar(15))

➤ **Mata\_kuliah**

- **Kode\_matakuliah** : kode untuk mata kuliah (varchar(15))
- **Nama\_matakuliah** : nama lengkap mata kuliah (varchar(100))

➤ **Ruang\_kelas**

- **Kode\_ruangkelas** : kode untuk ruang kelas (varchar(15))
- **Kapasitas\_ruangkelas** : kapasitas untuk ruang kelas (integer)

3) Menentukan relationship (hubungan) antar entitas

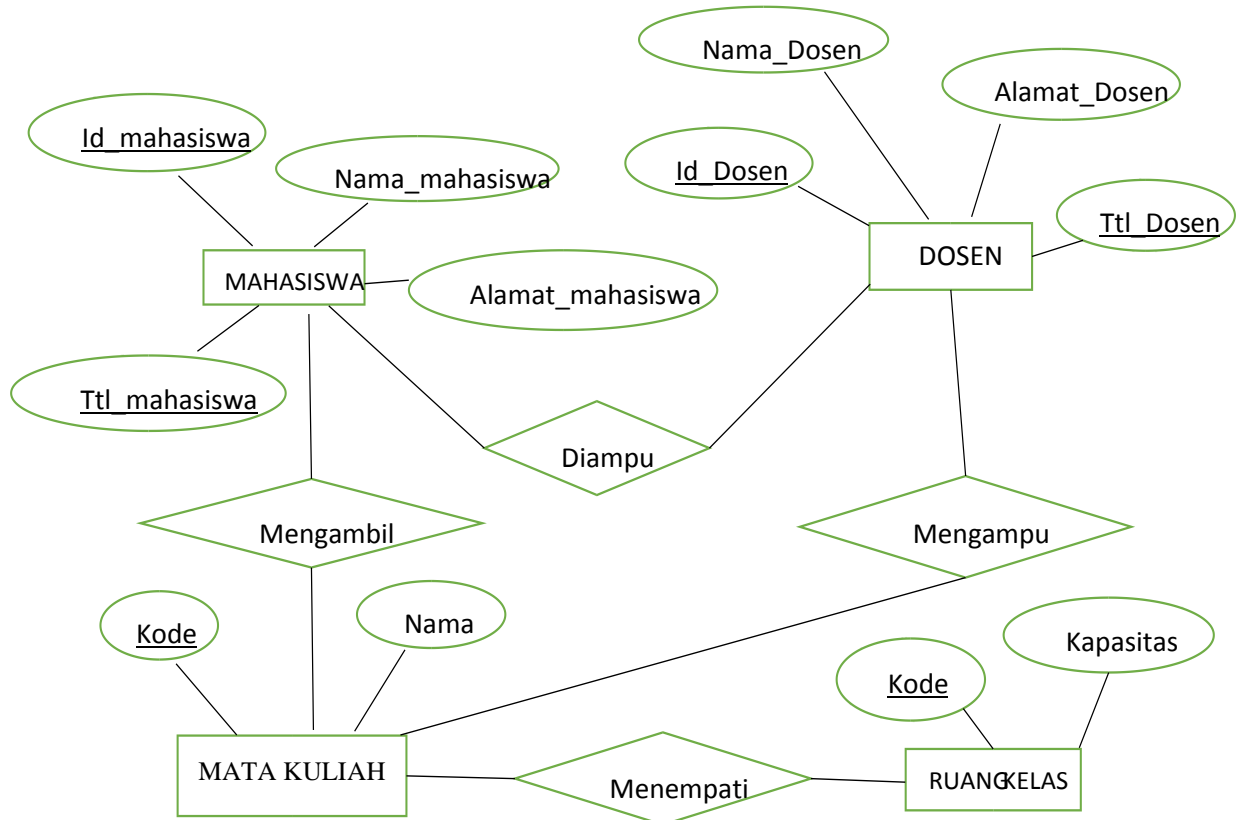
	<b>Mahasiswa</b>	<b>Dosen</b>	<b>Mata_kuliah</b>	<b>Ruang_kelas</b>
<b>Mahasiswa</b>	-	m : n	m : n	-
<b>Dosen</b>		-	m : n	m : n
<b>Mata_kuliah</b>			-	m : n
<b>Ruang_kelas</b>				-

**Hubungan :**

- **Mahasiswa mengambil Mata\_kuliah :**
  - **Tabel utama** : Mahasiswa, Mata\_kuliah
  - **Tabel kedua** : mahasiswa\_take\_matakuliah
  - **Relationship** : Many-to-many (m:n)
  - **Attribute penghubung** : Id\_mahasiswa, Kode\_matakuliah (FK Id\_mahasiswa, Kode\_matakuliah di **mahasiswa\_take\_matakuliah**)
- **Dosen mengajar Mahasiswa:**
  - **Tabel utama** : Dosen, Mahasiswa
  - **Tabel kedua** : dosen\_teach\_mahasiswa
  - **Relationship** : Many-to-many (m:n)
  - **Attribute penghubung** : Id\_mahasiswa, Kode\_dosen (FK Id\_mahasiswa, Kode\_dosen di **dosen\_teach\_mahasiswa**)
- **Dosen mengampu Mata\_kuliah:**
  - **Tabel utama** : Dosen, Mata\_kuliah

- **Tabel kedua** : dosen\_keepup\_matakuliah
- **Relationship** : many\_to\_many (m:n)
- **Attribute penghubung** : Kode\_dosen, Kode\_matakuliah (FK Kode\_dosen, Kode\_matakuliah di **dosen\_keepup\_matakuliah**)
- **Mata\_kuliah menempati Ruang\_kelas :**
  - **Tabel utama** : Mata\_kuliah, Ruang\_kelas
  - **Tabel kedua** : mata\_kuliah\_fill\_ruang\_kelas
  - **Relationship** : many-to-many(m:n)
  - **Attribute penghubung** : Kode\_matakuliah, Kode\_ruangkelas (FK Kode\_matakuliah, Kode\_ruangkelas di **mata\_kuliah\_fill\_ruang\_kelas**)
- **Dosen menggunakan Ruang\_kelas:**
  - **Tabel utama** : Dosen, Ruang\_kelas
  - **Tabel kedua** : dosen\_use\_ruang\_kelas
  - **Relationship** : many-to-many(m:n)
  - **Attribute penghubung** : Kode\_dosen, Kode\_ruangkelas (FK Kode\_dosen, Kode\_ruangkelas di **dosen\_use\_ruang\_kelas**)

#### 4) Menggambar ER Diagram



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.

**Jawab :**

- 1) Menentukan entities (object-object dasar) yang perlu ada di database.

- **anggota** : menyimpan semua data pribadi semua anggota perpustakaan
- **pegawai** : menyimpan semua data pribadi semua pegawai perpustakaan
- **buku** : menyimpan semua data buku di perpustakaan
- **denda** : menyimpan informasi tentang denda

- 2) Menentukan attributes (sifat-sifat) masing-masing entity sesuai kebutuhan database

- **anggota :**

- no\_anggota : nomor identitas untuk anggota perpustakaan (integer) PK
- nama : nama untuk semua anggota perpustakaan (varchar(45))
- alamat : alamat lengkap anggota perpustakaan (varchar(255))
- tgl\_lahir : tanggal lahir semua anggota perpustakaan (varchar(45))
- jurusan : jurusan semua anggota perpustakaan (varchar(45))

- **pegawai :**

- no\_pegawai : nomor identitas untuk pegawai perpustakaan (integer) PK
- nama : nama untuk semua pegawai perpustakaan (varchar(45))
- alamat : alamat lengkap pegawai perpustakaan (varchar(255))
- no\_tlp : nomor telp semua pegawai perpustakaan (integer)
- jabatan : jabatan dari semua pegawai perpustakaan (varchar(45))

- **buku :**

- no\_buku : nomor untuk buku perpustakaan (integer) PK
- judul : judul dari buku perpustakaan (varchar(45))
- pengarang : pengarang dari buku perpustakaan (varchar(45))
- thn\_terbit : tahun terbit dari buku perpustakaan (integer)
- penerbit : penerbit dari buku perpustakaan (varchar(45))

- **denda :**

- kode\_denda : kode denda dari perpustakaan (integer) PK
- tarif\_denda : tarif dari denda di perpustakaan (varchar(45))
- jenis\_denda : jenis denda di perpustakaan (varchar(45))
- tgl\_pinjam : tanggal peminjaman buku di perpustakaan (varchar(45))
- no\_anggota : nomor identitas untuk anggota perpustakaan (integer)

### 3) Menentukan relationship (hubungan) antar entitas

	<b>anggota</b>	<b>pegawai</b>	<b>buku</b>	<b>denda</b>
<b>anggota</b>	-	n:1	m:n	1:n
<b>pegawai</b>		-	m:n	-
<b>buku</b>			-	-
<b>denda</b>				-

### Hubungan

- **anggota meminjam buku**

- Tabel utama : anggota, buku
- Tabel kedua : anggota\_borrow\_buku
- Relationship : many-to-many (m:n)

- Attribute penghubung : no\_anggota, no\_buku (FK no\_anggota, no\_buku di anggota\_borrow\_buku)

- **anggota bayar denda**

- Tabel utama : anggota
- Tabel kedua : denda
- Relationship : one-to-many (1:n)
- Attribute penghubung : no\_anggota (FK no\_anggota di denda)

- **pegawai melayani anggota**

- Tabel utama : pegawai
- Tabel kedua : anggota
- Relationship : one-to-many (1:n)
- Attribute penghubung : no\_pegawai (FK no\_pegawai di anggota)

- **pegawai mendata buku**

- Tabel utama : pegawai, buku
- Tabel kedua : pegawai\_register\_buku
- Relationship : many-to-many (m:n)
- Attribute penghubung : no\_pegawai, no\_buku (FK no\_pegawai, no\_buku di pegawai\_register\_buku)

#### 4. Menggambar ER Diagram

